

Commodore
WORLD
Nº 15 - MAYO 1985
315 PTAS.
Joysticks: cómo usarlos en tus programas
Tus primeros pasos en Basic
C-16: Comandos e intérpretes
Expansor de líneas
Piratería de Soft: polémica.
NOVEDAD
PC de COMMODORE

BOXER 12

high resolution monochrome monitor 12"

NEW 85
NOVEDAD 85

ELECTRICAL ENVIRONMENTAL CHARACTERISTICS

CRT	SIZE	12"
	DEFL. ANGLE	90°
DISPLAY FORMAT	CHARACTERS	2000 (80 × 25)
VIDEO	INPUT SIGNAL	COMPOSITE VIDEO
	VIDEO SIGNAL	1 Vpp pos.
	RISE/FALL TIME	≤ 30 ns
	BANDWIDTH	20 MHz
	CENTRE RESOLUTION LINES/IN	1000
	INPUT RESISTANCE	75 Ohm
BLANKING TIME	HORIZONTAL	≤ 8 μs
	VERTICAL	≤ 700 μs
COMP. SYNC.	H. SYNC.	15.650-15.750 KHz
	V. SYNC.	50-60 Hz
EHT	(Ib = 0)	13 KV
POWER SUPPLY	INPUT VOLTAGE	min. 180 max. 264 Vac
	CONSUMPTION	30 VA
GEOMETRY	RASTER DISTORTION	max 1 %
	SCAN LINEARITY	max 10 %
	FOCUS	internal control
	V. AMPLITUDE	internal control
	V. FREQUENCY	internal control
	V. UPPER AND LOWER LINEARITY	internal control
	H. AMPLITUDE	internal control
	H. FREQUENCY	internal control
	H. LINEARITY	internal control
	H. PHASE	internal control
ENVIROMENTAL	AMBIENT TEMPERATURE	0° C + 40° C
	AMBIENT HUMIDITY (not condensed)	5-90 %
	STORAGE TEMPERATURE	40° C + 65° C
	STORAGE HUMIDITY (not condensed)	5-90 %
WEIGHT	GROSS/NET	5,7/6,6 Kg.

● *audio optional*

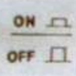
HANTAREX



BOXER 12



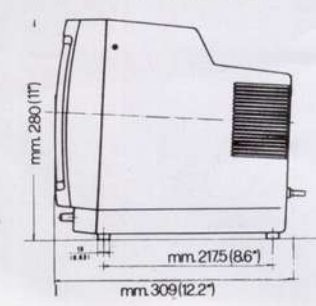


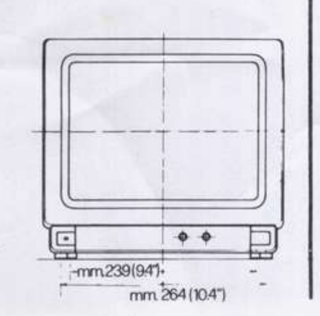


TRATTAMENTO SCHERMO: SCURO - ANTIRIFLETTENTE
SCREEN TREATMENT: DARK GLASS - ETCHED

FOSFORO - P31 - VERDE MEDIO-BREVE
PHOSPHOR - P31 - GREEN MEDIUM-SHORT

DATI MECCANICI
MECHANICAL DATA







HANTAREX®
Electronic Equipment Manufacturer
QUALITY . RELIABILITY . SERVICE

Aragón, 210, 1°, 1ª - Barcelona 11 - teléf. (93) 3232941 - telex 98017

Commodore WORLD

Commodore World está publicado por SIMSA y la colaboración de todos nuestros lectores

EQUIPO:
Manuel AMADO; Nieves CHESA; José Luis ERRAZQUIN; Lola HERMOSELL; Miguel A. HERMOSELL; Alvaro IBAÑEZ; María LOPEZ; Juan MARQUEZ; Pere MASATS; Cristóbal ORENES; Rafael PARDO; Angel RODRIGUEZ; Diego ROMERO; Albert SANGLAS; Jordi SASTRE; Valerie SHANKS; Francisco ZABALA... ...Y NUESTROS LECTORES

SIMSA
Coordinador María López
c/Barquillo, 21-3º Izda.
28004 Madrid. Teléf.: (91) 231 23 88/95
DELEGACION EN BARCELONA:
Sant Gervasi de Cassoles, 39 despacho 4.
08022 Barcelona
Tels.: (93) 212 73 45/212 88 48
Colabora MEC-COMMODORE con Club Commodore
Coordinador Pere Masats
Valencia, 49-53 - 08015 Barcelona
Teléf.: (93) 325 50 08

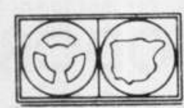


Imprime:
IBERDOS S.A.
Germán Pérez Carrasco, 24.
28027 Madrid
Depósito Legal: M-2944-1984

SUMARIO

PROXIMO NUMERO

- VIC-20 MULTICOLOR
- SERPIENTE DE LA MUERTE
- BASE DE DATOS (1ª PARTE)
- INICIAMOS UNA NUEVA SERIE SOBRE LOS SPRITES
- ... Y TODAS VUESTRAS COLABORACIONES



Commodore World es miembro de CW Communications/Inc., el grupo editorial más grande y de mayor prestigio en el mundo en lo que se refiere al ámbito informático. Dicho grupo tiene a su cargo la edición de 57 publicaciones relacionadas con los ordenadores en 20 países. Nueve millones de personas leen una o más publicaciones del grupo todos los meses. El grupo editorial está integrado por: **ALEMANIA** Computerwoche, Microcomputerwelt, PC Welt, Software Markt, CW Edition/Seminar, Computer Business, Run (Commodore), Apple's. **ARABIA SAUDI:** Saudi Computerworld. **ARGENTINA:** Computerworld-/Argentina. **ASIA:** Asian Computerworld. **AUSTRALIA:** Australia Computerworld, Macworld and Directories, PC World. **BRASIL:** Data News, MicroMundo. **DINAMARCA:** Computerworld/Danmark, PC World y Run (Commodore). **ESPAÑA:** Computerworld/España, MicroSistemas, Commodore World, PC World. **ESTADOS UNIDOS:** Computerworld, Hot CoCo, InCider, InfoWorld, PC World, 80-Micro, Mac World, Micro Market World, 73 magazine, Run (Commodore). Focus Publications. **FINLANDIA:** Mikro. **FRANCIA:** Le Monde Informatique, Golden (Apple), OPC (IBM) y distributique. **HOLANDA:** Computerworld Benelux, PC World Benelux. **ITALIA:** Computerworld Italia, PC Magazine. **JAPON:** Computerworld Japan. **MEXICO.** Computerworld/México, Compumundo. **NORUEGA:** Computerworld Norge, PC World y Run (Commodore). **REINO UNIDO DE GRAN BRETAÑA:** Computer Management, Pc Business World, Computer News, Computer Business Europe. **REPUBLICA POPULAR DE CHINA:** China Computerworld. **SINGAPUR:** Asian Computerworld. **SUECIA:** ComputerSweden. MikroDatorn. Svenska PC. **VENEZUELA:** Computerworld Venezuela.

SUPER-INTERESANTISIMO

PIRATERIA DE SOFT ...Y ALGO MAS

San Sebastián, 8 marzo 1985

Queridos amigos de COMMODORE WORLD:

Aunque con la presente no pretendo hacer una crítica al artículo aparecido en el número 13 de la revista titulado "PIRATERIA DE SOFT=ROBO", sí que quiero hacer unas puntualizaciones que supongo que no sólo me atañen a mí.

Los calificativos usados en el mencionado artículo son un poco duros, pero al fin y al cabo son los ún cos que la Real Academia de la Lengua permitiría usar en estos casos.

Aunque hay momentos en los que cualquiera de los que andamos en este mundillo podemos sentirnos un poco tocados, muchos como yo, hemos aprendido mucho, por no decir muchísimo, intentando copiar programas, que por mil caminos caen en nuestras manos. Una salvedad, en este momento poseo cerca de doscientos programas para el C-64 y os indico que de todos ellos, tan sólo ocho o diez han sido adquiridos por mí a la revista, el resto han sido algunos de mi cosecha propia, copiados de otros que me han prestado, traducidos, etcétera.

También, como salvedad, os indico que casi todos los programas que yo poseo han sido a su vez copiados por otros amigos, ya que formamos una especie de "club de intercambios", sin que por el momento, yo particularmente, haya recibido un solo duro por ello.

Existen en la actualidad multitud de programas en el ámbito internacional que por una causa o por otra no han llegado a España. Debido a las características particulares de mi trabajo, he de viajar mucho, por lo que tengo alguna facilidad en estar un poco al día de lo que existe en el mercado internacional, aplicable a mi ordenador.

¿Qué calificativo merecería para el autor del artículo, que por cierto no sé quien es, aquellas personas que compran un programa, se meten a fondo con y en él, lo traducen, lo adaptan a unas necesidades particulares, y luego...?

Me gustaría saber qué empresa de SOFT no hace o ha hecho eso, con mas o menos asiduidad. Como he indicado anteriormente, yo personalmente lo he hecho, pero no dispongo de una empresa de SOFT, por lo que cuando tengo un programa en mis manos, me meto a fondo con él, lo traduzco, si es necesario, lo adapto a mis necesidades, y luego no lo uso tan siquiera, copiándoselo al anteriormente mencionado pseudoclub de intercambios, por cierto nada formal al que se pueden agregar todas aquellas personas que lo deseen. Con este método hemos conseguido hacernos con un buen paquete de buenos programas sin tener que desembolsar los, algunas veces abusivos, precios que tienen en el mercado.

Es ilegal vender programas (C) sin conocimiento del autor, pero es tan ilegal como copiar una cinta de cassette y revenderla, cosa que en la actualidad es relativamente normal.

Sin embargo en el momento que haces el camino que he indicado anteriormente de buscar, comprar, meterte a fondo con él, traducirlo, adaptarlo, etc., deja en principio de ser ilegal, aunque moralmente lo sea.

Como me estoy alargando mucho y no es mi intención ser pesado, y a esta altura de la carta no sé si he dicho algo o no voy a hacer un resumen.

PREGUNTO: ¿Entramos dentro de la calificación dada por el autor del artículo personas como nosotros?, ¿es suficiente un artículo en una revista para sensibilizar a la gente a trabajar contra las empresas o particulares que venden lo que les cuesta X Pts. a X×X×X Pts.?, ¿qué medios tenemos los particulares para saber si un programa que venden las empresas de SOFT es o no "pirata"?

PROPONGO: a todos los usuarios la correspondiente denuncia a la revista y por parte de esta, su correspondiente espacio en la misma, reservado para éstas denuncias.

Nada más por el momento, espero no haberme extendido mucho, seguir adelante con vuestra línea y todos nos beneficiaremos.

Un saludo afectuoso.

A.J.O.G.

HOLA AMIGOS:

Os escribo la presente, para hacer unas cuantas puntualizaciones sobre el artículo aparecido en vuestra revista de marzo.

En principio, estoy de acuerdo con lo de que la venta de **soft** ilegal "fusilado" es un robo, y también estoy de acuerdo en que a la persona que hace un programa, le debe sentar bastante mal el ver su programa por la calle en copias "fusiladas" y además reconozco que es pisarle la idea, imaginación y trabajo que le haya costado hacerlo.

Una de las causas de esto, pienso que se debe en bastante culpa a los comerciantes legales del SOFT, pues pienso que venden los programas, a mi modo de ver, a un precio abusivo y esto lo digo sin saber los precios que estos comerciantes pagan por programas y cintas o discos, aunque creo que trabajarán con un margen bastante amplio.

También se podría hablar de la gente que hace programas como COPY 190 e IMITATOR, que también son programas como los demás y sin embargo sirven para "C O P I A R" otros programas —que conste que me parece muy bien que existan este tipo de programas.

Estoy también de acuerdo con lo de que no se debe pagar ni un duro por estas copias de programas fusilados, pues se pueden conseguir más tarde o más temprano por amistad o compañerismo, no por afán de lucro.

Otra cosa que me gustaría añadir, es que yo estaría dispuesto a CAMBIAR programas, **no a venderlos** con todos los que quieran ponerse en contacto conmigo.

Ya me despido esperando no haberme enrollado demasiado, pero no sin antes felicitaros por vuestra labor en esta revista.

Se despide un amigo

José Emilio Bases Muñoz
Avda. Santa Isabel, 6, bl. 7-2ºb. Bº Santa Isabel.
Telf.: 57 38 74. 50016-Zaragoza
Socio de la suscripción nº 5628

Reus, 18 de marzo de 1985

Hola artistas:

La presente es para discrepar con vosotros en vuestra decisión de suprimir las reseñas de intercambio de soft de la sección de mercadillo.

Pienso que es una decisión totalmente desafortunada y una grave falta de consideración hacia nosotros (vuestros lectores) que al fin de cuentas somos quienes os mantenemos y nos estáis haciendo pagar a todos, justos por pecadores.

Estoy muy de acuerdo en que el pirateo (venta ilegal) de soft, perjudica no sólo a las casas debidamente autorizadas que se dedican a su venta, sino a todos nosotros, tal y como correctamente razonáis en vuestra editorial del nº 13 (marzo), (sin ir más lejos yo mismo estudio programación); pero también soy consciente, Y ESTOY SEGURO DE QUE NO SOY EL UNICO QUE PIENSA ASI, de que los precios del soft en España están por las nubes (cualquier juguecito más de 2.000 ptas.). Y si no, explicarme: ¿Por qué en Andorra, los mismos juegos, mismo fabricante, idéntica procedencia, etc., rondan las 800 (ochocientas) ptas.? Sí, sí, ya sé, me diréis que es que allí no se pagan impuestos, pero... de 800 a 2.000 ptas. ¿no os parece que son muchos impuestos?, es un incremento del 150%. Todos sabemos que la presión fiscal está aumentando en este santo país, pero gracias a Dios todavía no hemos llegado a ese nivel. Entonces, ¿dónde va a parar esa diferencia?, creo que está bastante claro ¿no? Pienso que para ser comerciante hay que tener un mínimo de ética y no querer matar la gallina de los huevos de oro en dos días, porque luego pasa lo que pasa. Si me equivoco, seguro que saldrá alguien a aclarar ese agujero negro y nos explica dónde va esa diferencia o qué se hace con ella y obtendrá mis disculpas. Si no sale, ya sabéis, QUIEN CALLA, OTORGA.

Encima, resulta que vas a comprarte cualquier programa bueno (de ayuda al programador, procesador de textos, etc...), y aparte del precio (no por las nubes, sino por la ionosfera), te dan las instrucciones en inglés o en el mejor de los casos te "obsequian" con un minimanual que teóricamente es una traducción de las instrucciones originales. Ja, ja, que risa. ¿Tanto cuesta hacer una traducción bien?

¿Qué ocurre entonces? pues que tú compras el programa "A" y yo compro el programa "B". Hacemos un intercambio, y resultado: los dos tenemos el "A" y el "B". NO HAY OTRA SOLUCION ANTE ESOS PRECIOS ABUSIVOS. ¿ESO ES PIRATEO?, ¿ESO ES ROBO? Más bien creo que es un acto simbólico de ayuda mutua, se trata de un simple trueque (te cambio una cabra por una oveja) modalidad que como sabéis se utiliza desde muy antiguo. Y si no, ¿cómo podría un sólo usuario poseer una buena y completa "programoteca"?, pues de ninguna manera, o gastándose una fortuna.

Creo que es lo mismo intercambiar programas de ordenador, que cambiar sellos de Correos, que hacerlo con postales, monedas o novelas del oeste. **Lo importante es que no haya ni una sola peseta por medio.** ¿Cómo evitarlo? MUY SENCILLO: cuando recibimos una de esas propuestas de venta, lo único que tenemos que hacer es IGNORARLA. Estoy convencido de que NUESTRA INDIFERENCIA HARA DESISTIR A LOS AUTORES.

Por otro lado, comprendo vuestra posición ya que supongo que habréis recibido presiones por parte de los anunciantes (comerciantes) que como es lógico, tratan de evitar estos intercambios que les perjudican a ellos directamente y al fin y al cabo también ellos os mantienen.

Bueno, de una u otra forma, creo que sería muy positivo por vuestra parte (rectificar es de sabios) que reconsideraseis vuestra decisión. Bueno, pues se acabó, espero que la presente no acabe en la papelera y hasta la próxima, mientras tanto recibid un cordial saludo.

Os otorgo total libertad para publicar total o parcialmente el contenido de esta carta y al menos tendré el consuelo de no haber perdido el tiempo escribiéndola.

Juan Caballero Díaz
Bº Gaudí, Bl. C. 3ª nº 8
Reus (Tarragona)

Estimados amigos de C.W.: Soy suscriptor (nº 2192) de vuestra revista y gran aficionado a C-64, que me permite a mí, como a mi mujer e hijos pasar horas agradables en familia gracias a vuestra revista y en especial a la sección "Mercadillo" he contactado con muchos padres de familia, estudiantes, etc..., con la misma afición que yo por el C-64. También he contactado con algunos especialistas del pirateo ofreciéndome toda clase de juegos, aparatos, etc..., y os puedo decir que a esos listillos los he despreciado o mejor dicho ignorado. Asimismo también os puedo decir que he recibido innumerables ofertas de compra de juegos y utilidades a las cuales no he hecho el menor caso ya que mi profesión no tiene nada que ver con este mundo.

Os digo todo lo anterior para que comprendáis que los puramente aficionados como yo vamos a sentir enormemente que hayáis suprimido la sección de mercadillo dedicado a los intercambios y creo que no es justo que paguemos los "justos por los pecadores". Supongo que los anunciantes en vuestra revista os han obligado a ello.

Por otro lado y en defensa del llamado "pirateo casero" os debo decir que muchas empresas de Soft se merecen que se les copien, se les dañen económicamente, etc., etc..., ya que salvo honrosas excepciones el 99% de estas empresas se dedican únicamente a importar y tal como las reciben y sin incluir una mala documentación en español lo ponen a la venta en España. Yo he tenido la desgracia de comprar el Calc Result, Magic Desk, etc., en España y os puedo decir que no hay derecho con los precios ya que estos son hasta 4 y 5 veces su valor de origen en las tiendas en EE.UU. (estoy suscrito a Commodore Computer Gazzete de EE.UU. y a RUN de Alemania) y ya no digo nada si analizamos precios para Hardware.

Por todo lo anteriormente dicho creo que al suprimir los intercambios las empresas de Soft no van a mejorar las ventas puesto que cuando se tiene ya un poco de experiencia y poseyendo una tarjeta Diners, American Express, Visa, etc..., por dos perras gordas y por correo te envían de EE.UU. o Inglaterra cualquier juego en menos de quince días.

Lo que tienen que hacer esas empresas es dar calidad y precio. Todo lo demás viene detrás.

Benito Regidor
Avda. de Valladolid, 47-B 5º A. Telf.: 248 00 38
28008 Madrid

Hola, amigos de Commodore World:

No soy suscriptor de vuestra revista, pero sí asiduo lector.

Os envío un comentario para que si queréis lo publiquéis en la sección de "Carta Blanca".

CONCURSOS PERMANENTES

Colaboraciones de Programas y Magia

La verdad sea dicha: os estáis portando; las colaboraciones que nos están llegando son numerosas y de muy buena calidad. Por lo tanto, vamos a aclarar las condiciones de nuestros concursos permanentes.

1) Los sorteos se realizarán en los meses de junio y diciembre.

2) Se sortearán siempre ocho premios, cuatro de 15.000, 10.000, 5.000 y 3.000 pesetas en metálico de "Commodore World" y cuatro premios del mismo valor en material de Microelectrónica y Control, entre todas las colaboraciones publicadas.

3) Se sortearán siempre seis premios, tres de 3.000, 2.000 y 1.000 pesetas en metálico de "Commodore World" y tres del mismo valor en material de Microelectrónica y Control, entre todas las contribuciones publicadas en la sección de "Magia".

4) Adicionalmente, se sortearán diversos premios de cintas, juegos, objetos curiosos, etc., entre todas las colaboraciones.

5) Los autores de las colaboraciones vendidas dentro de nuestro "Servicio de Cintas" percibirán el 20% del precio de la cinta.

6) Todas las colaboraciones deben venir escritas a máquina y los programas grabados en cinta (si es posible), o con el listado completo en impresora. Nuestros lectores más jóvenes pueden escribir a mano pero con letra muy clara.

7) Quedan automáticamente descalificadas tanto del sorteo como del "Servicio de Cintas" las colaboraciones que hayan sido enviadas a otras revistas.

8) Las colaboraciones se enviarán a "Commodore World", c/Barquillo, 21-3º izda. 28004-Madrid.

Durante este verano he estado en Estados Unidos conviviendo con una familia. Yo, desde Navidad, ya tenía un Commodore-64, y siempre compraba vuestra revista: no tenía ni idea y todos los programas me parecían una gozada. Pero desde que llegué y compré la revista RUN me quedé decepcionado; era una réplica casi exacta de la versión española; los mismos apartados, las mismas gráficas, e incluso la caligrafía del listado era igual. Las únicas diferencias eran la expresión, la portada, y que no imprimáis todos los programas de esta revista, sólo algunos, pero poco a poco lo iréis haciendo (como el programa Word Search, de julio del 84, lo habéis publicado en Navidad). Ellos tampoco tenían concursos, ni participación de los lectores con programas, pero a pesar de eso, sigue siendo muy parecida a su colega estadounidense.

César Aguilar
c/ General Eguia nº 10
48010 Bilbao

Respuesta a las cartas anteriores

1) Ni por un momento se nos ocurrió atacar ni meternos con aficionados, usuarios o profesionales de la informática que "destripan" los programas para su propio uso o beneficio —esto es precisamente parte del juego del mundo de la informática, su aprendizaje, investigación, etc.—. Nosotros somos los primeros que no dejamos sano programa que cae en nuestras manos, y es incluso nuestra obligación.

2) Tampoco nos metemos, en absoluto, con el intercambio, tal cual, de programas sean del tipo que sean.

Una cosa es el intercambio y otra cosa es la *copia y venta lucrativa* y, no seamos ingenuos, *muy lucrativa*.

3) No deseamos ni vamos a hacer aquí una defensa particular de las empresas de SOFT, pero sí de la empresa de SOFT. Y, por favor, no me digáis que estoy intentando hacer sutilezas demagógicas. Las empresas de soft y la empresa del soft son dos temas muy distintos.

En las primeras las hay, como en cualquier negocio o profesión, serias, muy serias y nada serias, incluso piratas como el que más. La empresa del soft es toda la inventiva, creatividad, desarrollo y en muchos casos riesgo económico que todo avance de la humanidad ha supuesto o supone.

Aquí no hay diferencias entre la importante multinacional, y el más supermodesto individuo particular que trabaja en una habitación de su casa.

4) Nos habláis de las posibles presiones de los comerciantes que se anuncian. Sobre esto sí que deseo puntualizar que no somos simplemente una firma dedicada a publicaciones generales o especializadas. SIMSA es parte del grupo C.W. Communications, compañía internacional dedicada única y exclusivamente al *periodismo* informático. En el mundo del periodismo hay presiones inmensas procedentes de todas las esferas. Las que puedan ejercer anunciantes o lectores son simplemente una más. El periodismo serio y responsable a muy pocas presiones cede y C.W. Communications tiene fama a nivel internacional de no casarse con nadie y... más de un problema hemos tenido.

Y aquí sí quiero dejar patente que, efectivamente, si hemos tenido presiones, pero muy particularmente de anunciantes a los que nos negamos a anunciar precisamente por esta razón, por ser empresas piratas y, os puedo asegurar que están dispuestos a pagar *muy bien* los anuncios.

5) Estamos de acuerdo en que el soft en España es caro. Tampoco voy a meterme aquí, ni es nuestra misión, hacer un estudio de las diversas economías de las casas de soft, ni del sistema económico en general. Ideologías sobre este tema las hay para todos los gustos y aquí mismo, en la redacción, las tenemos de todos los colores.

Nosotros no nos dedicamos a la venta de soft. Nuestro soft es público y todo el mundo puede copiarlo dándole al teclado que es lo nuestro.

Solamente quiero hacer un par de comentarios generales con el único fin de aclarar ideas.

• Los profesionales del soft tienen toda una empresa que mantener, sueldos, gastos de mantenimiento, si son creadores larguísimas horas de trabajo, si no lo son altísimos copyrights, comisión de ventas a las tiendas vendedoras, promoción y publicidad del producto, etc., etc. Los porcentajes de beneficios o ganancias se van distribuyendo a lo largo de una larga cadena.

• El pirata, en un gran porcentaje casero y funcionando por medio de un apartado de correos, tiene una inversión de un programa copiador, cintas vírgenes para ordenador (50-60 ptas. en mayorista), un sello de correos para enviarlo y tres minutos de tiempo por copia.

(Pasa a página 57)

METEDURAS DE PATA

• En el número 14, el listado del programa de gráficos en alta resolución salió mal por la impresora. El símbolo extraño que aparece en la línea 20 es una flecha arriba "↑" y el +PI ⌉ que aparece en otras (la 200 por ejemplo) es simplemente PI (SHIFT flecha arriba).

• En MAGIA, el truco CAMPANA DE GAUSS lleva invertidos un paréntesis y un asterisco.

• En el Reductor de Programas del número 13, la línea 860 ha de ser: IF N< 2 THEN 930 y no "mayor que" como aparecía en el listado.

HACIA EL SUSCRIPTOR 8.000
RECORDATORIO

Sorteo entre TODOS los sucriptores de:
1) Viaje para dos personas a elegir entre París, Lisboa, Atenas, Amsterdam, Niza, Viena, Canarias, Mallorca.
2) Impresora donada por Microelectrónica y Control.
3) 80 programas valorados en 160.000 ptas. donados por la casa POKE, S.A. de Barcelona.

ULTIMO NUMERO
DE SUSCRIPTOR = 7.088

INDICE DE ANUNCIANTES

JOYSTICK

Cuando compré mi Commodore 64 dejé de lado sus usos más prácticos para dedicarme al estimulante, y al mismo tiempo desesperante, desafío de escribir juegos. Ocho semanas después había terminado mi creación.

"Vaya", comentó uno que decía ser mi amigo, con cara de poco convencido después de someterlo a prueba. "¿Quieres decir que ni siquiera utiliza un joystick?".

Tenía razón, aunque me costó admitirlo. Un juego manejado desde el teclado no es, en absoluto, tan divertido como uno que se juega con joystick. Desafortunadamente para nosotros, futuros diseñadores de juegos, no resulta nada fácil descifrar cómo se programan estos instrumentos tan sencillos.

Mi primer intento de introducir una rutina con joystick en un juego hizo que me esforzara en descifrar las técnicas que había descubierto en las últimas páginas de las revistas. Ecuaciones tales como:

JV = 15 — (PEEK (56321) AND 15)

casi me hicieron abandonar. Podía seguir lo que hacían los PEEKS solamente de una forma vaga, pero la extraña combinación con un AND era ya demasiado. Y, para colmo, estaba lleno de paréntesis por todas partes. El resultado consiguiente fue que, durante bastante tiempo realmente, procuré encarecidamente hacer que todos mis juegos fueran mediante el teclado.

¿A que te suena familiar? Pues entonces anímate. Lo que se expone a continuación es una guía del principiante para escribir programas referentes a los joysticks. El proceso gira en torno al comando Peek y al comando And.

No te dejes arrullar por el sonido de estas dos palabras pues son conceptos engañosos. Sin embargo, el aprender bien cómo utilizarles trae consigo recompensas que se extienden más allá de escribir código para joysticks. Son las instrucciones Basic más cercanas al lenguaje máquina (junto con Poke) y para mí resultaron realmente un punto de apoyo básico para adentrarme en semejante mundo.

Lo cómodo de hacer PEEKS con el joystick es que no tienes que calcular ningún número binario mayor que el 31.

Sobre el sistema binario

Antes de atacar esto, es importante tener en cuenta algunos conceptos básicos sobre la forma en que el ordenador considera los joysticks. Si tienes la Guía de Referencia de Programador del Commodore 64, puede que te hayas dado cuenta de que lo referente a los joysticks se incluye en el capítulo de Entradas/Salidas. Esto es debido a que efectivamente la lectura de las señales producidas por el joystick es realmente una forma más de entrada al ordenador.

Tanto si conectas como si no un joystick, el ordenador comprobará 60 veces al segundo si se ha introducido algún tipo de información a la máquina a través de este port. Almacena esta información en una de las 64.000 celdas de memoria. Cada byte (o celda) del ordenador tiene una dirección. La dirección para el joystick que se encuentra situado en el port 1 es la 56321. Para el que se encuentra en el port 2 es la 56320.

Te recomiendo que escribas estas direcciones en un sitio que tengas a mano, ya que te van a indicar constantemente qué es lo que está sucediendo con los joysticks.

Durante todo el tiempo en que no esté sucediendo nada con los joysticks, se encontrará un 1 en cada uno de los primeros cinco bits de los bytes de joysticks. El resultado de cinco 1,s es 31. Si escribes 31 en binario, aparecerá lo siguiente: 11111.

El bit que se encuentra situado más a la derecha corresponde a la dirección más alta del joystick. Tan pronto como tires hacia arriba del joystick, el dígito de la derecha se convertirá en el valor 0, y el valor de los últimos cinco bits de la posición 56321 (ó 56320) cambiará a 30 (11110 en binario).

El segundo bit de la derecha corresponde a un tirón del joystick hacia abajo. Puesto que si se encontrase que éste bit estuviera a 1 significaría que, al ser el segundo, representa un 2 en decimal, entonces un tirón hacia abajo, que cambia el bit a cero, cambiará el total del valor del byte a 29 (31 menos 2, ó 11101 en binario).

El tercer bit, con un valor decimal de cuatro, te indica si el joystick está siendo empujado hacia la izquierda (11011, ó 27) y el cuarto revela un empujón hacia la derecha (resultando por tanto una combinación binaria de 10111, ó 23 en decimal). Si el quinto bit se pusiera a 0, sería porque se ha pulsado el botón de fuego (botón de disparo).

Si la lógica binaria todavía no te parece que tenga mucha lógica, debes leer las páginas 75-78 de la Guía del Usuario. Aquí lo más importante a tener en cuenta es que los bytes 56320 y 56321 tienen 8 conmutadores representados por los 8 bits de cada byte, de los cuales los cinco situados al final del byte son los que indican qué es lo que está haciendo el joystick en cada momento. Si en cualquier posición de estos cinco bits aparece un cero, significa que el joystick está haciendo algo.

Deja de leer por un momento y teclea en el ordenador la rutina que se muestra a continuación y que, además de resultar sencilla, sirve para ver una aplicación de todo lo que se acaba de contar.

```
10 JV=31—(PEEK(56321) AND 31)
20 IF JV=1 THEN PRINT "ARRIBA"
30 IF JV=2 THEN PRINT "ABAJO"
40 IF JV=4 THEN PRINT
   "IZQUIERDA"
50 IF JV=8 THEN PRINT
   "DERECHA"
60 IF JV>15 THEN PRINT
   "FUEGO!"
70 GOTO10
READY.
```

S: COMO USARLOS EN TUS PROGRAMAS

Este artículo explica cómo tu ordenador maneja los joysticks, proporcionándote consejos y técnicas sobre la forma de usarlos en tus programas.

Haciendo Peeks

Ahora que ya has entendido la forma en la que el ordenador maneja los valores de los joysticks, el siguiente paso va a consistir en encontrar la manera en que pueden leerse dichos valores. El comando Peek va a hacer justamente eso; mediante un Peek se echa un vistazo al contenido de la posición de memoria que le indiques al ordenador. De esta forma, si tecleas Peek (56321), el ordenador va a tomar el valor que se encuentre correspondiente al joystick 1. (Por favor, fíjate en los paréntesis, totalmente necesarios, entre los que se inserta la dirección de la posición de memoria que se intenta investigar).

Sin embargo, si únicamente tecleas PEEK (56321), el ordenador olvidará inmediatamente qué es lo que ha visto. Por consiguiente, debes de asignar una variable a este valor. Por ejemplo:

J1 = PEEK (56321)

De esta manera, cada vez que pases a través de esta sentencia, se actualiza J1 con el valor actual del joystick 1.

Eso es bastante sencillo. Cuando entres en el mundo de los PEEKS más sofisticados que manejan sprites y gráficos, es cuando las cosas empiezan a ponerse realmente duras. Puesto que la lectura de los joysticks no tiene nada que ver con eso, podemos continuar.

El comando AND

Lo cómodo de hacer PEEKS con el joystick es que no tienes que calcular ningún número binario mayor que el 31. Esto me causó una gran confusión en mi primer intento de incorporar valores para el joystick dentro de mis rutinas. Para poder leer solamente los primeros cinco bits, tienes que separarlos de los otros tres bits que componen el byte. El lenguaje Basic y el código máquina ofrecen la misma técnica para realizar esta operación: el comando And.

Esta simple palabra, que se encuentra en todas partes, es uno de los conceptos más duros de aprender para muchos de los programadores que comienzan. Las frases que intentan explicar el concepto —operadores lógicos, tablas de verdad booleanas y máscaras de bit— no ayudan demasiado.

Para ilustrar la utilización del comando And, imagina que acabas de hacer un Peek en la posición 56321. Ahora vas a recubrir el byte de ocho bits con otro byte. Imagina que este nuevo byte es una fila de ocho ventanas que se superponen justamente encima del byte del joystick que acabas de encontrar mediante el Peek anterior. Si asignas a cada una de las ventanas un valor de 1, cada una de estas se abrirá y revelará el bit correspondiente que tenga debajo y que pertenece al byte del joystick.

Puesto que quieres ver únicamente los 5 últimos bits de la posición de memoria 56321, hay que abrir solamente las ventanas 0 — 4 del byte de recubrimiento. Esto hace que el valor sea 31. El comando And sitúa este byte como recubrimiento del valor de la posición 56321. Puesto que únicamente las ventanas 0—4 están ▶



JOYSTICKS

Como

se ha establecido

con anterioridad,

estos números

están constantemente

cambiando

cuando se conecta

el mando joystick,

incluso

si no los está

leyendo nadie.

abiertas, los único que lees son los bits 0—4; los bits 5, 6 y 7 son, por tanto, despreciados y puestos a cero. De esta manera, el valor que ves a través de este byte que hace de máscara estará comprendido entre 0 y 31.

Desde este punto de vista, es bastante simple la cuestión de descifrar la dirección a la cual está apuntando un joystick. Las páginas 13 y 14 de tu manual de referencia se refieren a esto.

A continuación se presenta una versión actualizada de la sentencia utilizada para el joystick.

J1 = PEEK (56321) AND 31

Ahora, cada vez que el bucle haga que pase el programa a través de esta sentencia, te indicará cuáles son los cinco bits más bajos de la posición 56321. Si, como ocurre en el caso más frecuente, únicamente necesitas comprobar la dirección del joystick y no del botón de disparo, puedes utilizar el valor 15, en lugar del valor 31, después del comando And, ya que ya no es necesario que te fijes en el bit que se encuentra en el cuarto lugar (con un valor de 16, por tanto). Cuando quieras comprobar únicamente el botón de disparo, utiliza

J1 = PEEK (56312) AND 16

Esta sentencia abre la ventana cuarta únicamente para desvelar el estado del botón de disparo.

Definiendo los valores del joystick

Tienes que aprender un último truco. Generalmente los programadores prefieren obtener una lectura inversa del valor del joystick. Por ejemplo,

J1 = PEEK (56321) AND 15

te devolverá un valor de 14 si el joystick está apuntando hacia arriba. Es más fácil pensar que realmente es esto si se indica con un 1 en lugar de un 0, puesto que así se ve que se relaciona directamente con el bit de la posición 0. Para conseguir que esto sea así, la sentencia debe alterarse para que quede como se indica a continuación:

J1 = 15 —(PEEK (56321) AND 15)

Esto tiene la conveniencia adicional de mantener el resultado dentro del margen que va desde 0 a 10, como puede apreciarse en la tabla adjunta.

Rápidamente, revisemos los conceptos anteriores con la ayuda de la tabla que debe estar a la vista: joystick arriba, el bit 0 se pone a 0; cuando abajo, el bit 1 se pone a 0; joystick a la izquierda, el bit 2 se pone a 0; joystick a la derecha, el bit 3 se pone a 0. Ahora veamos la tabla 1.

Fíjate, ¿qué es lo que está sucediendo aquí? ¿Qué tienen que ver esas diagonales? Fácil. Arriba y a la izquierda dejará activados a 1 únicamente los bits 1 y 3, generando un valor de 10 (15 menos 10 es 5). Arriba y a la derecha dejará activados únicamente los bits 1 y 2, proporcionando un valor de 6 (15 menos 6 es 9). Abajo y a la izquierda dejará activados los bits 0 y 3, que da un total de 9 (15 menos 9 es 6). Abajo y a la derecha dejará activados los bits 0 y 2, resultando en un valor de 5 (15 menos 5 es 10).

No es posible tener un valor que esté por debajo del 5. (Por ejemplo, un 4 significaría que el joystick estaba siendo empujado en cualquier dirección excepto a la izquierda, siendo esto una posibilidad que se encuentra más allá del objetivo de incluso el joystick más avezado. También fíjate, que los valores 12 y 8 no dan como resultado ninguna dirección.

Como se ha establecido con anterioridad, estos números están constantemente cambiando cuando se conecta el mando joystick, incluso si no los está leyendo nadie. Son actualizados 60 veces al segundo, frecuencia que suele ser suficiente para comprobar y llevar a cabo cualquier cosa que quieras. ¿Qué paso viene después de que se haya leído este valor? Utilizarlo. Intenta que funcione esta sencilla rutina que se da a continuación:

```
10 POKE 53281,0:PRINT"J":
   POKE53281,1
20 X=1490:J1=56321
30 POKEX,81
40 JV=15—(PEEK(J1)AND15)
50 IFJV=4THENX=X—1:
   POKEX+1,32
60 IFJV=THENX=X+1:
   POKEX—1,32
100 GOTO30
READY.
```

Si tecleaste todas las sentencias correctamente, debes ser capaz de manipular una bola de lado a lado de la pantalla con el joystick enchufado al port 1. Si, es verdad, es una bola borrosa como mucho, pero consigue demostrar lo simple que resulta utilizar un mando joystick. Cualquier posicionamiento direccional diferente de los de izquierda y derecha no van a ser leídos. Esto es debido a que si el resultado de la lectura del joystick, que se realiza en las líneas 40 y 50, no es un 4 ó un 8, entonces el programa no hace nada excepto volver al comienzo del bucle para realizar la comprobación una vez más.

Tabla 1.
Equivalencias de los valores resultantes de realizar un Peek del Joystick

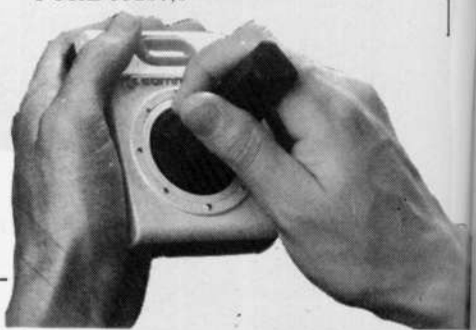
Valor del mando (56321)	15 menos el valor del mando	Dirección
15	0	ninguna
14	1	arriba
13	2	abajo
12	3	ninguna
11	4	izquierda
10	5	arriba-izquierda
9	6	abajo-izquierda
8	7	ninguna
7	8	derecha
6	9	arriba-derecha
5	10	abajo-derecha

Podrías hacer, con mucha facilidad, que las diagonales también te moviesen a la izquierda o a la derecha respectivamente, simplemente añadiendo dichas condiciones a las líneas 50 y 60. Por ejemplo, 50 podría leerse como:

IF JV=4 OR JV=6 THEN...

O podrías añadir una línea 70 de forma que quedase:

(IF PEEK(56321)AND16)=0 THEN POKE 53281,0



Contabilidad-64 así de fácil.

El programa Contabilidad 64 es líder en ventas, por su sencillez, rapidez, eficacia y precio.



Si precisa conocer todas las ventajas hoy, envíenos el cupón adjunto y recibirá gratuitamente el manual resumido de Contabilidad 64.

casa de software s.a.

c/. aragón, 272, 8.°, 6.ª
tel. 215 69 52
08007 barcelona

DESEO RECIBIR INFORMACION:
Nombre y dirección:

JOYSTICKS

Ahora, cuando pulsas el botón de disparo en cualquier momento, la pantalla se volverá negra. Básicamente, lo que está sucediendo aquí es que estás definiendo el joystick.

Ahora, cuando pulsas el botón de disparo en cualquier momento, la pantalla se volverá negra. Básicamente, lo que está sucediendo aquí es que estás definiendo el joystick. Puede causar el efecto que quieras. Puede incluso imprimir una palabra. Sólo tienes que modificar la línea 50 para que quede:

IF JV=1 THEN PRINT"OUCH!"

Ahora cualquier movimiento hacia arriba tendrá como resultado que la pantalla reaccione quejándose.

Alcanzando mayor sofisticación

Puedes y debes experimentar con unas cuantas rutinas sencillas como las que se han listado anteriormente para conseguir alcanzar una mejor comprensión de los conceptos que se encuentran involucrados en el tema. Sin embargo, encontrarás en el diseño de juegos, especialmente en Basic, que este tipo de rutinas es, con mucha frecuencia, demasiado lento, particularmente cuando tienes un buen montón de sentencias If... Then dentro del bucle. Existe una manera mejor, pero no es tan fácil de entender como la anterior.

Una rutina más sofisticada no puede incluir un puñado de sentencias del tipo IF joystick igual a esto THEN haz eso otro. En su lugar, el resultado de realizar un Peek al joystick tiene que alimentar directamente a una variable concreta cuya acción debe haberse definido ya con anterioridad. A continuación se expone cómo funciona esto.

Al principio del programa, lee las acciones que deseas que se lleven a cabo con cada movimiento del joystick para que se introduzcan en una variable cuyo valor interno sea el valor correspondiente al mando joystick. ¿Complicado? Entonces estudia el listado 1.

Esta rutina demuestra los valores suscritos del joystick con la ayuda de un sprite. Las líneas 10 y 20 cargan un array con los valores 0, 1 y -1. Las líneas 40—60 establecen los parámetros del sprite. La línea 80 lee el joystick y la línea 100 hace un Poke a las coordenadas X e Y de tu sprite con los valores del joystick. La línea 110 cierra el bucle para que se comience de nuevo otra vez.

Esta pequeña técnica es una joya. Date cuenta de que no hay ninguna sentencia If... Then. Se han evitado introduciendo los valores, proporcionados por el joystick, dentro de matrices con sentencias Poke. Por ejemplo, si se hubiera apuntado hacia arriba a tu joystick, la línea 80 habría asignado un 1 a JV. La línea 100 habría, por tanto, insertado ese 1 en las dos variables A( ) y B( ). Luego A(1) igual a -1. Se hace Poke V+1 (tu coordenada Y) consigo mismo más el uno negativo de A(1), y el sprite se desplaza un pixel más arriba en la pantalla. No se mueve diagonalmente porque B(1) es un 0, y puesto que añades eso a tu coordenada X en

POKE V, PEEK (V)+B(1)

no tiene lugar ningún movimiento horizontal.

Esta rutina, tal y como está, se ejecuta bastante rápidamente. Es una buena forma de permitir que se escriban juegos en Basic con una velocidad suficiente. Todos los 1s de la rutina representan el número de posiciones que quieres que tu sprite se mueva de una sola vez —creo que con tres es suficiente para que puedas verlo de una forma fácil. Incluso todavía, con la velocidad de esta técnica, pueden ejecutarse para realizar comprobaciones muchas sentencias If... Then, que buscan colisiones y otras condiciones típicas en los juegos, sin disminuir demasiado la velocidad.

Desde luego, si quieres realmente que se hagan las cosas de forma rápida, tienes que realizarlas en código máquina. Aunque el explicar tales procedimientos se encuentra más allá de la meta de este artículo, he incluido una rutina (ver listado 2) que puede utilizarse, sin ningún conocimiento de código máquina, para mover cosas de un lado a otro. No incluye el estado de disparo. Para utilizar esta rutina dentro de tus programas, incluye la línea 10 y las líneas Data en algún sitio al comienzo de tus programas y pon un SYS 49152 en el bucle principal de tu programa. El ejemplo que se muestra lo hace así. Estúdialo para entender cómo funciona.

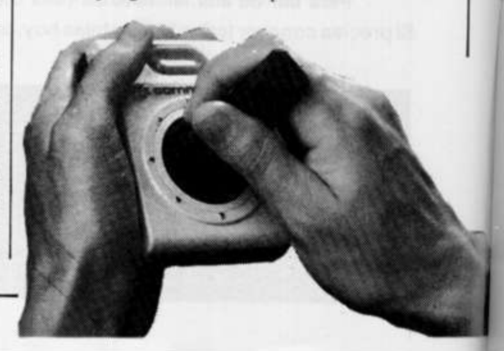
Si esto resulta demasiado complejo para tí, no te desanimes. Sólo tienes que intentar poner las rutinas como se listan aquí; llega a familiarizarte con ellas y pronto estarás utilizando los joysticks en tus juegos como un profesional.

LISTADO 1

```
10 FORT=1TO10:READA(T),B(T):
   NEXT
20 DATA-1,0,1,0,0,0,0,-1,-1,-1,1,-1,0,0,
   0,1,-1,1,1,1
40 V=53248:FORT=12288TO12350:
   POKET, 255: NEXT
50 POKE2040,192:POKEV,100:
   POKEV+1,100
60 POKEV+39,12:POKEV+21,11 :
   J1=56321
80 JV=15—(PEEK(J1)AND15)
100 POKEV,PEEK(V)+B(JV):
    POKEV+1,PEEK(V+1)+A(JV)
110 GOTO80
```

LISTADO 2

```
10 FORT=49152TO49204:READA:
   POKET,A:N EXT
40 FORT=12288TO12350:POKET,255:
   NEXT:V=53248
50 POKE2040,192:POKEV,100:
   POKEV+1,100
60 POKEV+39,12:POKEV+21,1:J1=
   56321
100 SYS49152
110 GOTO100
200 DATA173,1,220,41,15,201,14,240,15,
    201,13,240,18
210 DATA201,11,240,21,201,7,240,24,76,
    0,192,206,1,208
220 DATA206,1,208,96,238,1,208,238,1,
    208,96,206,0
230 DATA208,206,0,208,96,238,0,208,
    238,0,208,96,0
```



Por Michael BUCKLEY • Adaptado por Alvaro IBAÑEZ

NIMBOTS
VIC-20 - 8k

En Nimbots, tú y el ordenador —u otra persona que sea tu oponente— os vais turnando para quitar entre uno y cuatro Nimbots de acuerdo con ciertas reglas. Para mover, tienes que teclear las letras de los Nimbots que quieres quitar y pulsar después la tecla Return. Los Nimbots que quites de una sola vez deben estar situados en línea recta, ya sea horizontalmente o siguiendo cualquiera de las diagonales, pero no verticalmente, y además, no puede haber ningún espacio entre ellos. Por ejemplo, ADHL podría ser un movimiento que se consideraría legal, pero ADL no serviría —independientemente de que H esté situado o ya haya desaparecido de su sitio. (El ordenador no aceptará movimientos ilegales). El jugador que se vea obligado a realizar el último movimiento será el que pierda.

Pronto te enfrentarás con una docena de repugnantes Nimbots, cuya única obsesión es presidir tu derrota. Este ingenioso juego de la mente es, al mismo tiempo, divertido y desafiante

Ocho versiones

Las reglas antedichas describen la versión estándar. Cuando ya seas un maestro en esta versión, puedes intentar una variante según la cual el objetivo del juego es justamente lo contrario: tienes que intentar que seas tú el que tomes el último Nimbot.

Hay algunas versiones en las que se renuncia a la regla de no-espacios, es decir, todas las letras seguidas, de manera que, por ejemplo, el movimiento FHI podría ser un movimiento aceptable (a pesar de la ausencia de G). Sin embargo, la regla de seguir las líneas rectas se aplica a todas las versiones.

En cualquiera de las cuatro variantes pueden jugar una o dos personas. Esto te proporciona un total de ocho alternativas entre las que puedes elegir una,

Antepasados del Nimbots

Nimbots es uno de los muchos descendientes de Nim, un juego en el que dos jugadores se turnan en quitar una o más fichas de uno cualquiera de varios montones. El jugador que no puede hacer un movimiento es el perdedor, en otras palabras, el ganador es el que consigue coger la última ficha o grupo de fichas.

Después de jugar a este juego durante un rato, empiezas a reconocer ciertas posiciones "seguras" a partir de las cuales tu oponente es imposible que gane. Dos montones iguales son una posición segura: haga lo que haga tu oponente en un montón, tienes que hacer lo mismo en el otro. Por consiguiente, el jugar al Nim con tres montones, cada uno de los cuales contenga una, dos y tres fichas, respectivamente, es una posición segura ya que siempre puedes igualar el primer movimiento de tu oponente y forzar montones iguales.

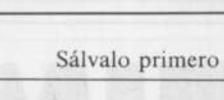
El VIC-20 es uno de los muchos microordenadores construidos en base a un microprocesador 6502. En el conjunto de instrucciones del 6502 se incluye la instrucción EOR, u "OR Exclusivo". La instrucción EOR compara dos números binarios, bit a bit, dando como resultando 0, cuando los respectivos bits son idénticos, y un 1 cuando encuentra que son diferentes.

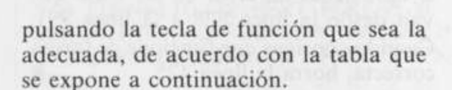
Por ejemplo, si se aplica la operación de Or exclusivo con los números 89 y 108, el resultado sería 53. Este ejemplo sirve para ilustrar a qué razón es debido el que se le denomine generalmente a esta instrucción "suma sin acarreo".

89 (decimal) = 1011001 (binario)
108 (decimal) = 1101100 (binario

53 (decimal) = 0110101 (binario)

Resulta obvio que aplicar la operación de Or exclusivo a un número consigo mismo va a tener como resultado un 0. De igual forma, la operación 1 EOR 2 EOR 3 va a tener como resultado un 0.





pulsando la tecla de función que sea la adecuada, de acuerdo con la tabla que se expone a continuación.

Función	Nº jugadors.	Espacios	Jugador Ultimo
F1	1	No	pierde
F3	1	No	gana
F5	1	Si	pierde
F7	1	Si	gana
F2	2	No	pierde
F4	2	No	gana
F6	2	Si	pierde
F8	2	Si	gana

En las versiones de un jugador, si no quieres hacer la jugada de apertura, para iniciar el juego, lo único que tienes que hacer es simplemente pulsar la tecla Return y el ordenador, de esta forma, jugará primero.

Cuando, al iniciar el juego, se te pide que lo digas, tienes que seleccionar el nivel de dificultad que quieras desde el cero hasta el nueve. En los niveles más altos, el ordenador juega impecablemente —pero tienes que tener en cuenta que puedes vencerle todavía, siempre que hagas todos los movimientos de forma correcta. En los niveles inferiores, el ordenador actúa a menudo de una forma aleatoria.

Sálvalo primero

Nimbots está escrito en Basic, con una subrutina de lenguaje máquina a la que se le hace POKE en el buffer del cassette y que comienza en la posición de memoria direccionada con 828. Para evitar la pérdida de un programa que no esté aún verificado ni probado, asegúrate de que lo salvas al menos una vez antes de que lo vayas a ejecutar.

El total de valores que puede haber en el array M% es de 49680 y la suma de todos los números a los que se les hace

Antes de que hubiese ordenadores, los matemáticos tenían otro nombre para esta operación: la denominaban la suma-nim. ¡La suma-nim de cualquier posición segura en el juego del Nim resulta que es 0!

Una versión de Nim comienza con tres montones de 3, 5 y 7 fichas. Puedes calcular la suma-nim y comprobar que el resultado es 1. Por ejemplo:

3 (decimal) = 011 (binario)
5 (decimal) = 101 (binario)
7 (decimal) = 111 (binario)

1 (decimal) = 001 (binario)

Tomando 1 de cualquier montón se reduce la suma-nim a 0. No puedes ganar si te enfrentas a situaciones tales como: 2, 5, 7, ó 3, 4, 7, ó 3, 5, 6 (a menos que tu contrincante cometa un error en alguna jugada posterior).

Ahora viene lo contrario. Normalmente este juego tiene establecido como objetivo justo lo inverso: tienes que intentar que el otro jugador tome la última ficha. La estrategia que debe seguirse con esta versión se le deja como ejercicio al lector.

(Pasa a pág. 16)

NIMBOTS

Poke en las posiciones de memoria 828-1003 es de 24627. Antes de que vayas a ejecutar el programa de Nimbots, tienes que introducir:

25 FORJ=0T071:T=T+M%(J):
NEXT:PRINTT:END

El programa debe presentar en pantalla el número 49680 y parar. Si se obtuviese cualquier otro número, se habrá producido un error en algún lugar situado entre las instrucciones Data que van de la 30 a la 100. Cuando consigas que se ejecute de manera totalmente correcta, reemplaza la línea 25 con la línea 125:

125 FORA=828T01003:T=T+PEEK(A):
NEXT:PRINTT:END

Esta vez, si no obtienes el número 24627, tienes un error en las Data que van desde la línea 828 a la línea 991. Cuando consigas que funcione de forma correcta, borra la línea 125. ■

```
10 DIMM%(71):PRINT"[CLR][7SPC]NIMBOT
S"
20 FORJ=0T071:READM%(J):NEXT:S2=3687
5:S3=S2+1:S5=S3+2:CL=S5+1:C1=255.5
30 DATA1,2,4,8,16,32,64,128,256,512,
1024,2048,3,6,24,48,96,384,768,3072
50 DATA17,34,68,136,272,544,1152,230
4,9,18,36,144,288,576,1280,2560
70 DATA7,56,112,896,273,546,1160,232
0,146,292,1312,2624,120,2321,1316
90 DATA5,40,72,80,130,257,260,514,64
0,1028,1032,1056,2049,2064,2112
100 DATA88,104,1060,1284,2065,2305
110 Y$="[HOM][23CRSRD]":X$="[22CRSRR]
"
120 DIMA%(4095):IFPEEK(828)-216THENF
ORA=828T01003:READP:POKEA,P:NEXT
130 PRINT"[2CRSRD][3SPC][BLU]PULSA[SPC]
UNA[SPC]TECLA[5SPC]PARA[SPC]ELEJIR[SPC]
VERSION:"
140 PRINT"[CRSRD][RVSON][SPC]JUGADOR
ES[SPC][SHIFT-]ESPA[SHIFT-]OL-[3SPC]
UNO[SPC][SHIFT-][SPC]DOS[SPC][SHIFT-]
CIOS[SHIFT-]TIMO[SPC]"
150 PRINT"[2SPC][RVSON]F1[RVSOFF][3SPC]
[RVSON]F2[RVSOFF][2SPC][SHIFT-][SPC]
NO[SPC][SHIFT-]PIER.[2SPC][RVSON]F3[RVSOFF]
[3SPC][RVSON]F4[RVSOFF][2SPC][SHIFT-]
[SPC]NO[SPC][SHIFT-]GANA[SPC]";
160 PRINT"[2SPC][RVSON]F5[RVSOFF][3SPC]
[RVSON]F6[RVSOFF][2SPC][SHIFT-][SPC]
SI[SPC][SHIFT-]PIER.[2SPC][RVSON]F7[RVSOFF]
[3SPC][RVSON]F8[RVSOFF][2SPC][SHIFT-]
[SPC]SI[SPC][SHIFT-]GANA"
170 GETA$:IFA$=""THEN170
180 F=ASC(A$)-132:Q=0:IFF<1ORF>8THEN
170
190 M=0:IFF>4THENF=F-4:Q=1:M=1:GOTO2
50
200 PRINT"[CRSRD][2SPC][BLU]ELIGE[SPC]
DIFICULTAD:":PRINT"[CRSRD][2SPC]DESD
E[SPC][RVSON]0[RVSOFF][SPC]=[SPC]FAC
IL"
220 PRINT"[CRSRD][2SPC]HASTA[SPC][RVSON]
9[RVSOFF][SPC]=[SPC]DIFICIL"
230 GETA$:IFA$=""THEN230
240 D=ASC(A$)-48:IFD<0ORD>9THEN230
250 P=8:IFF=1ORF=3THENP=25
260 POKECL,P:PRINT"[CLR]":P2=0:P3=0:
Y=1:X=1:C=31:IFF>1THENC=5:IFF>2THENC
=28:IFF=4THENC=158
280 B=4095:POKES2,0:POKES3,0:N=0:GOS
UB710:FORN=1T012:GOSUB710:Y=1:IFN>2T
HENY=5
```

Cómo funciona el programa

Puesto que hay 12 nimbots, y cada uno de ellos puede que esté o que no esté, hay 4.096 configuraciones posibles (2 elevado a la 12ª potencia). Cada uno de los elementos del array A% contiene un número que indica al ordenador qué movimiento tiene que hacer si encuentra la posición correspondiente. Si un elemento contiene un 0, entonces dicha posición es segura (para el contrincante) o es una posición que no puede analizarse (en la versión de baja dificultad), y el ordenador mueve de forma aleatoria.

Vamos a ver una muestra de cómo se puede jugar con un ejemplo en el que se supone que tenemos una situación en la pantalla en la que puede verse que A, F, J, K y L están visibles. A% (2119) representa esta situación. En binario, 2119 es 100001000111. Las letras del alfabeto A-L van de izquierda a derecha en este número binario. (Incluye los ceros para que el número de 12 dígitos resultante coincida con los 12 Nimbos alfabetizados). Si A% (2119) contiene 6, que se decodifica como 000000000110, entonces el ordenador seleccionará los Nimbots J y K, dejándote con tres Nimbots aislados (A, F y L) y una derrota inevitable, suponiendo que se está jugando según las reglas de la versión normalizada de juego sin espacios intermedios y que pierde la última persona que tiene que quitar un Nimbot.

¿Cómo se consigue que A% contenga estos valores?. Hay 72 movimientos legales, incluyendo todas las versiones, y se encuentran almacenados en el array M%. Por ejemplo, M% (13) quizá contenga, o puede que no, un 6 ya que este valor del array hace que se barajen todos los valores para conseguir que el juego sea aleatorio. En binario, 6 es 000000000110, que como ya se ha visto anteriormente, se refiere a los Nimbots J y K.

A% es muestreado desde el principio al final. Cuando se encuentra un elemento que contiene un cero, es decir, que representa una posición segura, entonces cada uno de los movimientos autorizados en M% que podrían conducir a esa posición se añaden al índice (seguro), proporcionando el índice de una posición no segura. El movimiento en M% entonces se almacena en cada una de las posiciones que el ordenador ha calculado que no son seguras.

Refiriéndonos al ejemplo expuesto antes, A% (2113) contiene un 0 —luego es una posición segura. Hay muchos movimientos en M% que podrían conducir a esa posición, uno de los cuales, en M% (13), es un 6. Sumando 6 a 2113 da un resultado de 2119. De esta manera, se llega a almacenar un 6 en M% (2119).

Desde luego, todo lo que se ha expuesto se realiza antes de que realices el primer movimiento. Es un procedimiento que lleva unos 20 minutos en lenguaje BASIC, sin embargo, solamente dura aproximadamente un par de segundos con la

```
290 IFN>5THENY=9:IFN>9THENY=13
300 X=N*6:IFN>2THENX=(N*24-Y*21+45)/
4
310 PRINTLEFT$(Y$,Y+1)LEFT$(X$,X);:P
RINTCHR$(X)"[RVSOFF][SHIFTM][COMMP][SHIFTN]
[CRSRD][3CRSRL][RVSON][COMMD]-[COMMF]
[CRSRD][3CRSRL][RVSOFF][COMMJ][RVSON]
[SPC][RVSOFF][COMML][CRSRD][3CRSRL][COMMV]
[SPC][COMMC][CRSRD][3CRSRL][COMML][COMMY]
[COMMJ]"
330 PRINTLEFT$(Y$,Y+4)LEFT$(X$,X+1)"
[RVSON]"CHR$(N+64):POKES5,0:NEXT:POK
ES5,15
340 FORN=1TO222:NEXT:POKES5,0
350 PRINT"[3CRSRD][SPC]UN[SPC]MOMENT
O[SPC]POR[SPC]FAVOR":H=50:IFF>2THENH
=71
370 POKE982,(H+1)*2:POKE856,1:IFF=2O
RF=4THENPOKE856,0
390 POKE1000,D*5/3:SYS828:POKE956,44
8-PEEK(956)
395 Q=-Q:W$="JUGADOR"+STR$(2+(Q=1)):
IFQ=0THENW$=""
396 PRINT"[HOM][19CRSRD][3SPC]";W$
400 INPUT"[HOM][20CRSRD][2SPC]MOV[M]
ENTO[10SPC][10CRSRL]";Q$:IFQ$<>""THE
N430
410 IFB=4095THEN500
420 GOTO400
430 V=0:FORN=1TOLEN(Q$):E=76-ASC(Q$)
:IFE>=0ANDE<12THENV=VOR(2↑E)
440 Q$=RIGHT$(Q$,LEN(Q$)-1):NEXT:FOR
J=0TOH:IFV=M%(J)THEN460
450 NEXT:GOTO400
460 IF(VANDB)<>VTHEN400
470 B=B-V:IFB=0THENONFGOTO610,580,61
0,580
480 FORN=1TO12:F=2↑(12-N):IFFANDVTHE
NGOSUB660
490 NEXT
500 IFMTHEN395
510 V=A%(B):IFVTHENFORN=0TO333:NEXT:
GOTO550
520 FORJ=0TO4:R=INT(RND(0)*51):V=M%(
R):M%(R)=M%(J):M%(J)=V:NEXT
530 FORJ=0TOH:V=M%(J):IF(VANDB)=VTHE
N550
540 NEXT:GOTO580
550 FORN=1TO12:F=2↑(12-N):IFFANDVTHE
NGOSUB660
560 NEXT:B=B-V:IFBTHEN400
570 IFF=2ORF=4THEN610
580 PRINT"[CLR]":FORT=1TO8:N=INT(T*1
2/7):PRINTLEFT$(Y$,(10-T)*2)"[2CRSRR]
[RVSON][BLK]>>>[SPC]GANASTE![SPC]<<
<"
600 GOSUB710:POKECL,(11-T)*9:NEXT:GO
TO640
610 FORT=13TO6STEP-1:N=INT(T*12/7):P
RINT"[CLR]"LEFT$(Y$,(14-T)*2)"[2CRSRR]
[BLK]>>>[SPC]PERDISTE[SPC]<<<"
630 GOSUB710:POKECL,(T-5)*27:NEXT
640 FORN=0TO666:NEXT:POKES5,0:FORN=0
TO666:NEXT:RUN
660 GOSUB710:Y=1:IFN>2THENY=5:IFN>5T
HENY=9:IFN>9THENY=13
670 X=N*6:IFN>2THENX=(N*24-Y*21+45)/
4
680 PRINTLEFT$(Y$,Y+1)LEFT$(X$,X);:P
RINTCHR$(C)"[3SPC][CRSRD][3CRSRL][3SPC]
[CRSRD][3CRSRL][3SPC][CRSRD][3CRSRL]
[3SPC][CRSRD][3CRSRL][3SPC]"
700 PRINTLEFT$(Y$,Y+4)LEFT$(X$,X+1)C
HR$(N+64):POKE36878,0:RETURN
710 POKES5,15:P2=C1-2↑(5.75-N/12):P3
=C1-2↑(6-N/12):POKES2,P2:POKES3,P3:R
ETURN
828 DATA216,24,165,47,105,7,133,163,
165,48,105,0,133,164
842 DATA24,165,47,105,158,133,165,16
5,48,105,0,133,166
855 DATA169,1,133,167,169,0,133,168,
165,167,10,133,169,165,168,42,133,17
0
873 DATA24,165,165,101,169,133,171,1
65,166,101,170,133,172
886 DATA160,1,177,171,208,93,136,240
,249,160,0
897 DATA177,163,133,174,200,177,163,
133,173
906 DATA165,167,37,173,208,68,165,16
8,37,174,208,62,208,199
```

subrutina que se incluye en lenguaje máquina. Si quieres comparar las dos versiones de dicha subrutina, el listado 2 muestra el código ensamblado junto a las correspondientes sentencias Basic en los campos de comentários.

Consecuencias

A continuación se van a citar algunas preguntas que me tuve que contestar antes de poder convertir estas pocas líneas de Basic en lenguaje máquina. La referencia al listado 2 te ayudará a entender las respuestas; se supone que se poseen algunos conocimientos del ensamblador 6502.

1.—¿Dónde se encuentran algunas posiciones seguras en la página cero para direccionamiento indirecto? Nimbots utiliza las posiciones 163-176. He utilizado este área sin ningún efecto perjudicial durante mucho tiempo, pero tienes que comprobar tu mapa de memoria para estar completamente seguro de que el uso que el sistema hace de estas posiciones de memoria no entra en conflicto con la utilización que tú le vas a dar.

2.—¿Cómo puede uno arreglárselas para manejar arrays en Basic con instrucciones de lenguaje máquina? Utiliza el vector comienzo de Arrays situado en las posiciones 47 y 48. Este apunta al inicio del primer array dimensionado — el array realmente es siete bytes más largo. Otros arrays aparecen ordenadamente a continuación, cada uno después de los siete bytes antedichos. En las líneas 829 - 853 del listado en ensamblador, puede verse cómo he almacenado la dirección del primer byte de M%, en las sentencias 163-164 y de A% en las líneas 165-166.

3.—¿Cómo puede mantenerse la relocabilidad cuando se necesita saltar más de 127 bytes? Utiliza bifurcaciones como medio de llegar hasta allí. Mira las líneas 916 y 918. Ambas son instrucciones BNE. Es obvio que la segunda no va a ser ejecutada nunca bajo circunstancias normales, únicamente está como instrucción sin efecto. Ahora fíjate en la línea 1001. Me gustaría haber puesto BNE 863 aquí, pero está más allá del rango permitido en direccionamientos relativos. Así que, pongo BNE 918, y en la instruccion 918, inserto el BNE 863 justo detrás de otro BNE ■

NIMBOTS



```
920 DATA24,165,167,101,173,133,175,1
65,168,101,174,133,176,6,175,38,176
937 DATA24,165,165,101,175,133,175,1
65,166,101,176,133,176,152,170
952 DATA160,0,177,162,240,9,177,175,
208,16,200,177,175,208,11
```

```
967 DATA160,1,165,173,145,175,136,16
5,174,145,175,138,168
980 DATA200,192,102,208,168,230,167,
208,2,230,168
991 DATA165,167,201,255,208,177,165,
168,201,15,208,171,96
```

```
.,  033C  D8        CLD

.,  033D  18        CLC
.,  033E  A5 2F     LDA $2F
.,  0340  69 07     ADC #$07
.,  0342  85 A3     STA $A3
.,  0344  A5 30     LDA $30
.,  0346  69 00     ADC #$00
.,  0348  85 A4     STA $A4

.,  034A  18        CLC
.,  034B  A5 2F     LDA $2F
.,  034D  69 9E     ADC #$9E
.,  034F  85 A5     STA $A5
.,  0351  A5 30     LDA $30
.,  0353  69 00     ADC #$00
.,  0355  85 A6     STA $A6

.,  0357  A9 01     LDA #$01      ; I=1
.,  0359  85 A7     STA $A7
.,  035B  A9 00     LDA #$00
.,  035D  85 A8     STA $A8

.,  035F  A5 A7     LDA $A7
.,  0361  0A        ASL
.,  0362  85 A9     STA $A9
.,  0364  A5 A8     LDA $A8
.,  0366  2A        ROL
.,  0367  85 AA     STA $AA

.,  0369  18        CLC
.,  036A  A5 A5     LDA $A5
.,  036C  65 A9     ADC $A9
.,  036E  85 AB     STA $AB
.,  0370  A5 A6     LDA $A6
.,  0372  65 AA     ADC $AA
.,  0374  85 AC     STA $AC

.,  0376  A0 01     LDY #$01      ;IF A%(I) <> 0 THEN 03D9
.,  0378  B1 AB     LDA ($AB),Y
.,  037A  D0 5D     BNE $03D9
.,  037C  88        DEY
.,  037D  F0 F9     BEQ $037A

.,  037F  A0 00     LDY #$00      ;J=0

.,  0381  B1 A3     LDA ($A3),Y   ;K=M%(J)
.,  0383  85 AE     STA $AE
.,  0385  C8        INY
.,  0386  B1 A3     LDA ($A3),Y
.,  0388  85 AD     STA $AD

.,  038A  A5 A7     LDA $A7       ;IF (I AND K) <> 0
.,  038C  25 AD     AND $AD       ;THEN 03D4
```

AMSTRAD
CPC-464
"Lo increíble"
Confirmado por la prensa especializada

tu Micro
Micro, cassette y monitor en plena armonía. Su Basic es el más rápido de su categoría, superando al del Commodore, al del BBC e incluso al del Sinclair.

Computer Schau
Usuarios y técnicos lo confirman: se ofrece una relación precio/prestaciones que parece imposible.

Computer persönlich
Por un precio sorprendente se ofrece algo increíble. Un Basic superlativo.

micro
No hay en el mercado ningún ordenador en este nivel de precio que pueda enfrentarse a él.

C'T
¡Solución total a un precio fenomenal!

POPULAR Computing WEEKLY
Un ordenador personal extraordinario con unas enormes posibilidades como ordenador de gestión.

Personal Computer World
Su Basic es rápido, más rápido que casi todos los basics de 8 bits y que algunos de 16 bits.

micro bit
Su Basic se puede considerar impresionante... tiene unas características no usuales en microordenadores de su categoría.

COMPUTER CHOICE
El mejor Basic que he visto.

SCIENCE VIE MICRO
Se asedia a los distribuidores para conseguir un AMSTRAD. La demanda es desbordante.

micros
Calificado de "increíble", las pruebas realizadas así lo han confirmado en casi todos los aspectos... es un equipo con posibilidades fuera de lo común...



AMSTRAD "Lo increíble"

NIMBOTS

```
.. 038E  D0 44     BNE $03D4
.. 0390  A5 A8     LDA $A8
.. 0392  25 AE     AND $AE
.. 0394  D0 3E     BNE $03D4

.. 0396  D0 C7     BNE $035F
.. 0398  18        CLC            ;S=I+K
.. 0399  A5 A7     LDA $A7
.. 039B  65 AD     ADC $AD
.. 039D  85 AF     STA $AF
.. 039F  A5 A8     LDA $A8
.. 03A1  65 AE     ADC $AE
.. 03A3  85 B0     STA $B0

.. 03A5  06 AF     ASL $AF
.. 03A7  26 B0     ROL $B0

.. 03A9  18        CLC
.. 03AA  A5 A5     LDA $A5
.. 03AC  65 AF     ADC $AF
.. 03AE  85 AF     STA $AF
.. 03B0  A5 A6     LDA $A6
.. 03B2  65 B0     ADC $B0
.. 03B4  85 B0     STA $B0

.. 03B6  98        TYA
.. 03B7  AA        TAX

.. 03B8  A0 00     LDY #$00       ;IF PEEK(PEEK(162))
.. 03BA  B1 A2     LDA ($A2),Y ; = 0 THEN 03C7
.. 03BC  F0 09     BEQ $03C7

.. 03BE  B1 AF     LDA ($AF),Y ;IF A%(S) <> 0 THEN 03D2
.. 03C0  D0 10     BNE $03D2
.. 03C2  C8        INY
.. 03C3  B1 AF     LDA ($AF),Y
.. 03C5  D0 0B     BNE $03D2

.. 03C7  A0 01     LDY #$01       ;A%(S) = K
.. 03C9  A5 AD     LDA $AD
.. 03CB  91 AF     STA ($AF),Y
.. 03CD  88        DEY
.. 03CE  A5 AE     LDA $AE
.. 03D0  91 AF     STA ($AF),Y

.. 03D2  8A        TXA
.. 03D3  A8        TAY

.. 03D4  C8        INY            ;J=J+1

.. 03D5  C0 66     CPY #$66       ;IF J<51 THEN 0381
.. 03D7  D0 A8     BNE $0381

.. 03D9  E6 A7     INC $A7        ;I=I+1
.. 03DB  D0 02     BNE $03DF
.. 03DD  E6 A8     INC $A8

.. 03DF  A5 A7     LDA $A7        ;IF I<4095 THEN 035F
.. 03E1  C9 FF     CMP #$FF
.. 03E3  D0 B1     BNE $0396
.. 03F5  A5 A8     LDA $A8
.. 03E7  C9 0F     CMP #$0F
.. 03E9  D0 AB     BNE $0396

.. 03EB  60        RTS            ;RETURN
```

Sandra

Texto y gráficos para el Commodore 64

Este paquete responde a un nuevo concepto de bases de datos, que si bien no es totalmente nuevo en informática lo es para el C-64 en nuestro país.

Se trata de un sistema de elaboración, almacenamiento y utilización de ficheros gráficos, y no es un simple programa que permite dibujar, pues fue diseñado para ser utilizado como una base de datos gráfica que permite a los profesionales de distintos campos mantener su fichero de trabajo en cinta o disco con ayuda de su ordenador.

El Sandra permite escribir con caracteres definidos por el usuario que son almacenados dentro de una base de datos gráfica, estos caracteres son denominados "robots gráficos" en el manual y, en la cinta o disco que acompaña al cartucho, podemos encontrar algunos ejemplos, entre ellos un alfabeto "manuscrito" con caracteres españoles incluidos (la eñe, apertura de interrogación, etc.) El número máximo de estos caracteres que podemos almacenar en cada fichero es de 128, pero nada impide utilizar varios ficheros para una sola hoja gráfica. Esto último se puede hacer cargando en primer lugar uno de los ficheros de "robots", pasándolos a las posiciones que deberán ocupar en la pantalla, y cargando posteriormente otro grupo de robots, transfiriéndolos a pantalla, y así sucesivamente.

Cada grupo de caracteres o robots pueden diseñarse para una aplicación determinada (medicina, arquitectura, electrónica, etc.). En la cinta de demostración podemos encontrar: manuscrito, minialfabeto y electrónica, y nosotros podremos crear todos los que necesitemos.

Cada hoja de Sandra está compuesta por 400 * 320 puntos, lo que permite almacenar el equivalente a 50 * 40 caracteres, y cada robot (carácter definido), puede tener un máximo de 4 * 3 caracteres o 32 * 24 puntos o pixels. Al igual que los ficheros gráficos, las hojas gráficas pueden almacenarse en cinta o disco. Cada hoja ocupa 73 bloques en el disco, y cada fichero de robots gráficos 38.

Quizás el tamaño de los ficheros y de las hojas pueda alarmar a los que no dispongan de unidad de disco, pero pueden tranquilizarse, ya que Sandra graba en cinta a la misma velocidad que la unidad de disco (utilizando un sistema turbo), y este sistema de almacenamiento rápido nos ha demostrado en las pruebas que no causa problemas de carga.

Otra peculiaridad de Sandra es que, al contrario que el modo habitual de funcionamiento del C-64, permite simultanear en pantalla todos los símbolos de que disponemos en el teclado. Así pues, podemos teclear texto o anotaciones en las hojas gráficas en mayúsculas, minúsculas y con los gráficos de las teclas Commodore y Shift, sin que al pulsar estas dos teclas cambie ninguno de los signos que ya estaban en la pantalla.

Debemos aclarar que naturalmente no podemos ver la hoja completa en la pantalla, tanto en el modo de texto como en el gráfico. Esto es natural si tenemos en cuenta que la máxima resolución alcanzable con el 64 es de 320 * 200 Puntos, mientras cada hoja gráfica de Sandra duplica esta capacidad. El modo de ver el resto de la hoja es moverse por medio de las teclas de cursor o el joystick en todas direcciones.

Una vez que hemos terminado de preparar la hoja con el esquema o gráfico que nos interesa podemos almacenarlo para su posterior utilización y/o imprimirlo en papel pulsando la tecla F1.

El equipo necesario para Sandra 64 es:

Ordenador Commodore 64 ó SX-64, cartucho Sandra, unidad de disco Commodore 1541 o Datassette, Impresora MPS 801 ó 1525, y naturalmente un monitor o televisor.

Los que no dispongan de la impresora Commodore, deberán consultar con Casa de Software en Barcelona si su impresora es compatible, ya que existe un gran número de impresoras centronics que funcionan con el Sandra (por ejemplo la Seikosha 550 A), pero la lista sería enorme. Tampoco se puede decir "todas las impresoras sirven", en concreto la mía no va bien (se trata de una impresora matricial marca Admate DP-80). Quizás sea fácil realizar los cambios para que funcione correctamente, pero no lo he probado. Con la impresora MPS 801 de la oficina toda va perfectamente.

Las teclas de función son las que nos ayudan a realizar todas las operaciones. En el menú principal sus funciones son:

F1 - Cambia cassette/disco.
F3 - Editar hoja gráfica.
F5 - Editar robot de trazo.
F7 - Ver el directorio del disco.

Si estamos editando la hoja gráfica, las funciones son:

F1 - Volcado a impresora.
F2 - Modo robot automático.
F3 - Modo gráfico.
F4 - Modo robot inverso.
F5 - Robot de trazo.
F6 - Cambia de hoja gráfica a robot y viceversa.
F7 - Cambia la velocidad de trazo.
F8 - Graba la hoja en cinta o disco.

Los modos "robot" sirven para transferir cualquiera de los símbolos definidos en el archivo utilizando a la posición donde se encuentra el cursor.

Mediante el modo gráfico podemos dibujar en la hoja con ayuda del joystick conectado al Port 2. La velocidad de desplazamiento del puntero por la pantalla podemos seleccionarla con ayuda de la tecla de función F7.

El modo de "robot inverso" permite que los dibujos realizados en los modos anteriores se vean aunque se superpongan, ya que el dibujo va apareciendo en el color contrario al del fondo de cada punto, de modo que si dibujamos una recta que cruce sobre un dibujo anterior, la línea aparecerá negra sobre el fondo blanco, y blanca sobre la zona ocupada por el otro dibujo.

En la cinta de demostración encontraremos ejemplos llamados: Estación, dentista, carta/-reyes, esquema y esquema 2.

Los ejemplos pueden servir para dar idea de las posibilidades, en los campos profesionales y amateurs de este programa.

Siendo este un sistema de gestión de ficheros gráficos, podría verse complementado con un buen sistema de gestión de bases de datos similar a la Super Base. Hemos preguntado sobre esta posibilidad a sus autores y sonrieron... Parece ser que de momento no disponen de este programa, pero tampoco desmienten la posibilidad de su aparición en un futuro próximo, ya que los datos almacenados por Sandra pueden ser transferidos a otros programas con ayuda de algún utilitario.



Conociendo los programas de que dispone esta casa, debemos suponer que este sistema de base de datos se encuentre en fase de preparación, pero su futura comercialización dependería de la calidad final del producto desarrollado, ya que no han querido comercializar algunos productos por no dar el nivel de calidad y versatilidad que ellos mismos se exigían.

El precio del Sandra es 19.000 ptas, y quien desee más información puede dirigirse a la casa que lo ha comercializado, sus señas son:

Casa de Software
c/Aragón, 272, 8, 6
Tel. 93 - 215 69 52
08007 Barcelona

NOVEDAD EN INFORMAT

El PC de Commodore, aparato magnífico y más económico del mercado

Ya está aquí. ¡Y menudo PC! Ha sido la estrella de Informat igual que fue estrella en Las Vegas en enero y lo acaba de ser en Hannover, la principal feria europea de Informática.

Y no es para menos y se lo merece —con una compatibilidad del 100 por 100, prestaciones del más alto nivel y el magnífico precio con que sale— 419.000 la unidad de base de 256K RAM y que incluye teclado, monitor y 2 unidades de discos— lo convierte en el PC más económico del mercado. No podemos olvidar tampoco su aspecto físico, una sobriedad de líneas que lo hace muy atractivo y elegante.

El commodoriano que esté pensando en ampliar su campo de acción difícilmente va a encontrar mejor solución que este PC.

No hemos tenido tiempo de probarlo a fondo dada la premura del cierre de la revista, pero para el próximo número, que dispondremos de uno de ellos (D. Carlos Domenech, presidente de MEC, nos ha prometido enviarnos uno inmediatamente), podremos dedicarle una amplia reseña. De momento sólo hemos podido ver en la feria de Barcelona, Informat, que las tarjetas de ampliación para gráficos en color y la tarjeta controladora de disco duro funcionaban muy bien, pues tuvimos uno de ellos funcionando con gráficos y otro conectado a un disco duro de 40 megabytes.

También vimos el Wordstar, la D-Base 3, el Lotus 1-2-3 y otros paquetes que funcionaban perfectamente en el Commodore PC.

La impresión que tuvimos fue totalmente satisfactoria, y aunque no pudimos comprobar a fondo el ordenador, sí hicimos correr en él algunos de los programas que llevábamos en el maletín (eran programas de un IBM-XT), y con gran satisfacción disfrutamos viendo la gran estabilidad y resolución de la pantalla, comprobando el suave y preciso movimiento de las teclas al pulsarlas (el típico de Commodore), sus eficaces y silenciosos drives y sus infinitas posibilidades. Tanto la estabilidad de la pantalla como el teclado nos parecieron muy superiores a los del IBM PC.

Entre sus características principales son destacables en la versión base podemos citar:

—256K RAM de 9 bits.

—Zócalos incorporados para ampliarlo hasta 640K.

—Dos unidades de disco incorporadas de 360K cada una.

—Monitor de fósforo verde de alta resolución.

—Interfaces serie y paralelo incluidos (RS-232 y centronics).

—Cinco slots de ampliación compatibles con todas las tarjetas del IBM PC.

—Alta resolución ya incluida.

—Y como ya hemos dicho, su precio es el mejor de los ordenadores de esta categoría, pues en las 419.000 Ptas. van incluidas todas las características anteriores.

Opcionalmente se le puede incorporar un disco duro de 10 Megabytes (lo incorpora el modelo PC-20).

Al utilizar el interface centronics para las impresoras podremos utilizar cualquiera de las del mercado que dispongan de este modo de trabajo, y todo aquel que conozca las impresoras disponibles, sabrá que aquellas con interface centronics son las más baratas, rápidas y con mejor relación calidad/precio.

Como disponemos de otro canal de comunicaciones de tipo serie con protocolo RS-232, podremos conectar un modem u otro tipo de impresora.

Si tenemos en cuenta la gran competitividad en precios y calidades que está consiguiendo Commodore en todo el mundo, podremos imaginar que este modelo puede hacer la vida difícil a muchas marcas que actualmente comercializan equipos PC-Compatibles.

Las especificaciones del nuevo ordenador Commodore PC-10 figuran en la siguiente tabla:

MICROPROCESADOR:
- Intel 8088. Procesador de 16 bits.
- Intel 8087. Procesador coma flotante (opcional).

FRECUENCIA DE RELOJ:
- Memoria: 4.77 Mhz.
- Principal 256K RAM (ampliable hasta 640K)
- Vídeo 32K RAM
- ROM 8K ó 16K

PANTALLA:
- Monitor fósforo verde monocromo 12", o
- Monitor color RGBI 12" (media resolución)
- Formatos pantalla:
  - Alfanumérico monocromo 80×25.
  - Gráfico monocromo 640×200 ó 640×352 pixels.
  - Alfanumérico color, 16 colores, 40×25 ó 80×25.
  - Gráfico color, 16 colores, 160×200 ó 320×200 pixels.
  - Gráfico color, 4 colores, 320×200 ó 640×200 pixels.
- Atributos alfanuméricos:
  - Alta intensidad, vídeo en negativo, destelleo, subrayado (sólo monocromo).

TECLADO:
- Separable.
- 84 teclas, incluyendo 10 teclas de función.

INTERFACES:
- Port Paralelo - Centronics.
- Port serie - RS232.
- Teclado.
- RGBI
- Monocromo compuesto.
- 5 slots de expansión (para PCBs compatibles PC).

ALMACENAMIENTO:
- Unidad doble de discos flexibles de doble cara de 360K de capacidad cada uno, o un floppy de 360K y un disco duro de 10 Mbytes.

SISTEMA OPERATIVO:
- DOS 2.11

BIOS: Rom de 8K para rutinas de E/S, manejo de interrupciones y bootstrap.

DOS: Carga automática desde disco. Manejo de ficheros e interpretador de comandos. Versiones traducidas para cada idioma.



Descubre las nuevas impresoras star para tu Commodore

Star ha lanzado al mercado los nuevos modelos de impresoras para Commodore; doble interface, paralelo centronics y Commodore.

Con este interface para Commodore logramos poder trabajar con todas las grandes prestaciones de esta gama de impresoras, es decir, una gran velocidad, fricción tracción, cinta de máquina de escribir, 100 tipos de letras diferentes... Además de todas estas prestaciones lograrás con tu Commodore transcribir todos estos tipos de letra con sus propios caracteres gráficos.

Si quieres hacer gráficos trabajando con el Simon's Basic, lograrás hacer Hard copys directamente de pantalla.

Todo esto y mucho más lograrás con las impresoras STAR.

De venta en establecimientos especializados:

IMPORTADO POR:



COMPONENTES ELECTRONICOS, S.A.

08009 BARCELONA. Consejo de Ciento, 409
Tel. (93) 231 59 13

28020 MADRID. Comandante Zorita, 13
Tels. (91) 233 00 94 - 233 09 24

Por Gerald CODDINGTON (RUN EE.UU.)
Traducido y adaptado por Alvaro IBAÑEZ

PERSECUCION

¿Cuántas veces has jugado a perseguir a tus amigos en lugares cerrados? He aquí un programa en BASIC que requiere dos joysticks, para que podáis jugar un amigo y tú a perseguiros el uno al otro a través de la pantalla



Antes de cargar el programa "Persecución", asegúrate de que los dos joysticks estén conectados al ordenador. La pantalla de presentación te ofrece la opción de leer las instrucciones si no estás familiarizado con el juego.

El objeto de esta "Persecución" es acumular dos minutos en el marcador sin haber sido cogido. Tu marcador aumenta cuando le toca perseguir a tu oponente.

Date cuenta de que hay dos lugares en la pantalla, arriba y abajo, en los que puedes esconderte. Usalos para tomar ventaja.

Aquí debajo está el desensamblado del programa que he usado. Si no lo entiendes no te preocupes. Diviértete jugando.

```
49152 sei
49153 lda   789
49156 cmp   #234
49158 bne   49173
49160 lda   #33
49162 sta   788
49165 lda   #192
49167 sta   789
49170 jmp   49183
49173 lda   #49
49175 sta   788
49178 lda   #234
49180 sta   789
49183 cli
49184 rts
49185 (aquí va la rutina en código
      máquina).
```

¡Recuerda el JMP 59953 al final! y no hagas POKES a las direcciones 788 ó 789 desde Basic.

NOTA: Si no dispones de dos joysticks, usa el que tengas en el Port 2, y simula el del Port 1 con las teclas:
"flecha a la izquierda" = izquierda
"2" = derecha
"1" = arriba
"CTRL" = abajo.

```
1 REM PROGRAMA PERSECUCION. POR:GERA
LD CODDINGTON. RUN EE.UU.
10 PRINTCHR$(142):PRINTCHR$(8):POKE5
3280,0:POKE53281,0:SR=1:GOTO90
20 POKEV+16,16:POKEV+43,FF:POKEV+8,4
0:POKEV+9,200
30 PRINT"[HOM][7CRSRD]"TAB(36)"P[CRSRL]
[CRSRD]E[CRSRL][CRSRD]R[CRSRL][CRSRD]
S[CRSRL][CRSRD]E[CRSRL][CRSRD]G[CRSRL]
[CRSRD]U[CRSRL][CRSRD]I[CRSRL][CRSRD]
D[CRSRL][CRSRD]O[CRSRL][CRSRD]R"
50 PRINT"[HOM][33SPC][CRSRD][CRSRR]T
IEMPO[5CRSRL][2CRSRD]";
55 PRINTRIGHT$(TI$,3)
60 IFH1$>"000200"ORH2$>"000200"THEN6
20
70 FORL=54272TO54296:POKEL,0:NEXT:S=
54272
80 POKES+24,15:POKES+19,138:POKES+15
,XX:POKES+18,17:POKEV+30,0:RETURN
90 PRINT"[CLR]":FF=5:POKE53280,254:T
I$="000000":H1$="000000":H2$="000000
"
100 V=53248:X=15:Y=47:Z=255:POKEV+30
,0
110 POKEV+21,19
120 POKE2040,13:POKE2041,13:POKE2044
,13
130 POKEV+23,255:POKEV+29,255
140 FORN=0TO62:READQ:POKE832+N,Q:NEX
T:GOTO900
150 POKEV+39,2:POKEV+40,5
160 IFSR=1THEN310
170 POKEV+0,255:POKEV+1,215:POKEV+2,
15:POKEV+3,47:POKE53280,5
180 FORI=1TO25:PRINT"[34CRSRR][RVSON]
[COMM5][SPC]":NEXT:Q=215
190 GOSUB20
200 POKE2018,160
210 IFFF=2THENH1$=TI$
220 PRINT"[HOM][33SPC][CRSRD][CRSRR]
TIEMPO[5CRSRL][2CRSRD][GRN]";
225 PRINTRIGHT$(H1$,3)
230 IFH1$>"00199"ORH2$>"000199"THEN6
20
240 IFPEEK(V+30)AND3=3ANDFF=5THENFF=
2:XX=30:TI$=H1$:POKE53280,2:GOSUB20
250 IFFF=5THENH2$=TI$
260 PRINT"[RED]"
270 PRINT"[HOM][33SPC][CRSRD][CRSRR]
TIEMPO[5CRSRL][4CRSRD]";
275 PRINTRIGHT$(H2$,3)
280 IFPEEK(V+30)AND3=3ANDFF=2THENFF=
5:XX=68:TI$=H2$:POKE53280,5:GOSUB20
290 IFH1$>"000199"ORH2$>"000199"THEN
620
300 GOTO210
310 V=53248:POKEV+30,0
320 PRINT
330 PRINT"[CLR][2CRSRD][WHT][31SPC]/

340 PRINT"[4SPC][SHIFTO][SHIFTP][SPC]
[SHIFTO][COMMY][SPC][SHIFTO][SHIFTP]
[SPC][SHIFTO][COMMY][SPC][SHIFTO][COMMY]
[SPC][SHIFTO][COMMY][SPC][COMMG][COMMN]
[SPC][SHIFTO][COMMY][SPC][COMMN][SPC]
[SHIFTO][SHIFTP][COMMN][SHIFTM][SPC]
[COMMH]
350 PRINT"[4SPC][SHIFTO][COMMY][SPC]
[SHIFTO][COMMY][SPC][SHIFTL][SHIFTN]
[SPC][COMMY][SHIFTP][SPC][SHIFTO][COMMY]
[SPC][COMMH][2SPC][COMMG][COMMN][SPC]
[COMMH][2SPC][COMMN][SPC][COMMH][COMMN]
[COMMM][SPC][SHIFTM][COMMH]
360 PRINT"[4SPC][COMMH][2SPC][SHIFTL]
[COMMP][SPC][COMMH][SHIFTM][SPC][COMMP]
[SHIFT@][SPC][SHIFTL][COMMP][SPC][SHIFTL]
[COMMP][SPC][SHIFTL][SHIFT@][SPC][SHIFTL]
```




```
[COMMP][SPC][COMMN][SPC][SHIFTL][SHIFT@]
[COMMN][2SPC][COMMH]
370 PRINT"[WHT][CRSRD][10SPC]POR[SPC]
GERALD[2SPC]CODDINGTON"
380 PRINT"[3CRSRD]"TAB(4)"QUIERES[SPC]
INSTRUCCIONES?[SHIFT SPC](S/N)"
390 AA=-35:FORI=35TO245
400 POKEV+0,I:POKEV+1,180
410 POKEV+2,I-AA:POKEV+3,180
420 GETA$:IFA$=""THEN450
430 IFA$="N"THENRESTORE:CLR:GOTO90
440 IFA$="S"THEN770
450 AA=AA+.07:POKEV+30,0
460 IFPEEK(V+30)AND3=3THENPRINT"[HOM]
[15CRSRD]"TAB(22)"TE[SPC]COGI!":GOTO
480
470 NEXT
480 FORI=1TO1000:NEXT:PRINTTAB(11)"[2CRSRU]
SI,[SPC]PERO[SPC]AHORA[SPC]ME[SPC]TO
CA[SPC]A[SPC]MI!!"
490 AA=-30:FORI=1TO1800:NEXT:PRINT"[2CRSRU]
[40SPC]"
500 PRINT"[CRSRU][36SPC]"
510 FORI=245TO35STEP-1
520 POKEV+0,I+AA:POKEV+1,180
530 POKEV+2,I:POKEV+3,180
540 GETA$:IFA$=""THEN570
550 IFA$="N"THENRESTORE:CLR:GOTO90
560 IFA$="Y"THEN770
570 POKEV+30,0
580 IFPEEK(V+30)AND3=3THENPRINTTAB(9
)"[CRSRU]LO[SPC]HICISTE!":GOTO610
590 AA=AA+.06
600 NEXT
610 FORI=1TO1000:NEXT:PRINT"[CRSRU][38SPC]
[CRSRU]":RUN310
620 FORI=VTOV+9:POKEI,0:NEXT:PRINT"[CLR]
[14SPC][COMM5]FIN[SPC]DE[SPC]JUEGO"
630 PRINT"[3CRSRD]TIEMPOS:[2CRSRD]":
PRINT"[SPC]JUGADOR[SPC]1[SPC]:[SPC]"
H1$:PRINT"[2CRSRD][COMM3][SPC]JUGADO
R[SPC]2[SPC]:[SPC]"H2$
640 HS=1:IFH2$>H1$THENHS=2
650 PRINT"[2CRSRD][WHT][2SPC]EL[SPC]
JUGADOR"HS"GANO!"
660 PRINT"[9CRSRD][5CRSRR]PULSA[SPC]
BOTON[SPC]EN[SPC]EL[SPC]JOYSTICK[SPC]
1[CRSRL][CRSRU]"
670 WAIT56321,16,16:CLR:RESTORE:GOTO
90
680 END
690 RESTORE:CLR:GOTO90
700 DATA0,0,0,0,126,0,0,255,0
710 DATA1,255,128,3,255,192,3,255,19
2
720 DATA3,153,192,3,153,192,3,153,19
2
730 DATA3,255,192,1,255,128,0,255,0
740 DATA0,255,0,0,66,0,0,66,0
750 DATA0,66,0,0,231,0,0,0,0
760 DATA0,0,0,0,0,0,0,0,0
770 FORI=VTOV+9:POKEI,0:NEXT:PRINT"[CLR]
[WHT]"TAB(12)"[WHT]--PERSECUCION--"
780 PRINT"[CRSRD][COMM3][SPC]EL[SPC]
OBJETIVO[SPC]DEL[SPC]JUEGO[SPC]ES[SPC]
NO[SPC]SER"
790 PRINT"[SPC]CAPTURADO[SPC]DURANTE
[SPC]DOS[SPC]MINUTOS.[SPC]EL"
795 PRINT"[SPC]PRIMERO[SPC]EN[SPC]CO
NSEGUIRLO[SPC]ES[SPC]EL[SPC]GANADOR"

800 PRINT"[CRSRD][COMM4][SPC]PARA[SPC]
ATRAPAR[SPC]A[SPC]LA[SPC]OTRA[SPC]PE
RSONA,[SPC]TIENES"
810 PRINT"[SPC]QUE[SPC]TOCARLE."
820 PRINT"[SPC]CUANDO[SPC]LO[SPC]HAC
ES,[SPC]TU[SPC]MARCADOR[SPC]EMPIEZA[SPC]
A"
825 PRINT"[SPC]CONTAR.[SPC]SI[SPC]LA
[SPC]OTRA[SPC]PERSONA[SPC]TE[SPC]ATR
APA,"
830 PRINT"[SPC]TU[SPC]CONTADOR[SPC]S
E[SPC]PARA[SPC]Y[SPC]EL[SPC]SUYO[2SPC]
AVANZA."
840 PRINT"[SPC]ESTO[SPC]CONTINUA[SPC]
HASTA[SPC]QUE[SPC]ALGUNO[SPC]DE[SPC]
LOS[SHIFT SPC][2SPC]JUGADORES[SPC]LL
EGA[SPC]A[SPC]DOS[SPC]MINUTOS"
850 PRINT"[COMM6][CRSRD][SPC]ESTE[SPC]
JUEGO[SPC]NECESITA[SPC]DOS[SPC]JOYST
ICKS,[6SPC]UNO[SPC]PARA[SPC]CADA[SPC]
HOMBRE."
860 PRINT"[COMM7][CRSRD][SPC]SI[SPC]
OLVIDAS[SPC]A[SPC]QUIEN[SPC]LE[SPC]T
OCA[SPC]PERSEGUIR,"
870 PRINT"[SPC]MIRA[SPC]EL[SPC]BORDE
[SPC]DE[SPC]LA[SPC]PANTALLA.":PRINT"
[CRSRD][9SPC]BUENA[SPC]SURTE[SPC]A[SPC]
AMBOS[SPC]!!!"
880 PRINT"[WHT][CRSRD][4SPC][RVSON]P
ULSA[SPC]BOTON[SPC]EN[SPC]EL[SPC]JOY
STICK[SPC]1"
890 GOTO670
900 FORADRES=49152TO49305:READDATTA:
POKEADRES,DATTA:NEXTADRES
909 REM ** DATAS PARA LENGUAJE MAQUI
NA **
910 DATA120,173,21,3,201,234
920 DATA208,13,169,33,141,20
930 DATA3,169,192,141,21,3
940 DATA76,31,192,169,49,141
950 DATA20,3,169,234,141,21
960 DATA3,88,96,173,0,220
970 DATA201,123,240,15,201,119
980 DATA240,20,201,126,240,25
990 DATA201,125,240,30,76,94
1000 DATA192,206,0,208,206,0
1010 DATA208,76,94,192,238,0
1020 DATA208,238,0,208,76,94
1030 DATA192,206,1,208,206,1
1040 DATA208,76,94,192,238,1
1050 DATA208,238,1,208,238,7
1060 DATA208,76,94,192,173,1
1070 DATA220,201,251,240,15,201
1080 DATA247,240,20,201,254,240
1090 DATA25,201,253,240,30,76
1100 DATA49,234,206,2,208,206
1110 DATA2,208,76,49,234,238
1120 DATA2,208,238,2,208,76
1130 DATA49,234,206,3,208,206
1140 DATA3,208,76,49,234,238
1150 DATA3,208,238,3,208,76
1160 DATA49,234,96,0,0,0
1170 IFPEEK(789)=234THENSYS49152
1180 GOTO150
```

Por Robert W. BAKER y Adaptado por Alvaro IBAÑEZ

C-64 Y VIC-20

EXPANSOR DE LIN

¿Están las líneas de tu programa sobrecargadas con demasiadas sentencias? Este programa rompe y acorta dichas líneas, haciéndolas más sencillas de editar.

En el Expansor de Líneas, al igual que ocurría con el programa "Reductor de Líneas" (Nº 13 "Commodore World" - Pág. 8), este programa fue escrito para el PET de Commodore. Ahora lo he actualizado para que funcione en el C-64 y el VIC-20.

"Expansor", un compañero del "Reductor", lee un programa Basic que se haya salvado en disco y crea una copia nueva, descompactada o expandida. Lleva a cabo esta tarea tomando cualquiera de las líneas que contiene sentencias múltiples (sentencias separadas por dos puntos) dividiéndolas en líneas de programa independientes con una nueva numeración de línea. De esta forma, pueden editarse las largas líneas que crea el programa "Reductor" para luego volver a compactar el programa.

Cuando las líneas multisentencia se dividen en varias, los números de las nuevas líneas se crean incrementando en 1 el número de la línea original por cada sentencia que se independiza. Este procedimiento se utiliza hasta que el número de una de las líneas que se están creando alcanza el número que originariamente tenía la siguiente línea del programa. Cuando esto ocurre, el resto de la línea original se copia como parte de la última línea generada, conservando los correspondientes signos de dos puntos donde sea apropiado situarlos, para separar las diferentes sentencias.

El programa debe tener en cuenta determinadas palabras clave, o "tokens", puesto que determinan si existe la posibilidad o no de que pueda ser dividida, en líneas independientes, una línea determinada del programa. De esta manera, cualquiera de los datos que van detrás de una de las instrucciones tales como GOTO, End, Run, If, Return, REM, Stop, List o CONT se copian, sin modificar nada, hasta el final de la línea actual del programa. Asimismo, una vez detectadas unas comillas en un programa, la línea tiene que ser copiada hasta llegar a otras comillas o se alcance el final de la línea.

Descripción de la línea

Cuando ejecutas el programa "Expansor", puedes controlar el tamaño de las líneas del programa que queremos que sean descompactadas.

El primer mensaje de entrada (línea 370) pedirá la longitud mínima de la línea que tiene que ser expandida. Esta longitud debe ser un número positivo con un valor situado entre 1 y 255, pero no se realiza ninguna comprobación del valor que se introduce. Si simplemente pulsas la tecla Return cuando se realiza esta petición, el valor por defecto forzará al programa a intentar dividir todas las líneas simples, cuando esto sea posible. El hecho de seleccionar un número, como el 20 por ejemplo, hará que las líneas pequeñas (con 20 o menos caracteres en dicha línea) permanezcan intactas, mientras que las líneas más largas sean expandidas.

El programa debe tener en cuenta determinadas palabras clave, o "tokens".

Después de que selecciones la longitud mínima de la línea que desees, entonces se te pide que proporciones el nombre del fichero donde se encuentra el programa en Basic que va a ser descompactado (líneas 390-400). El programa debe encontrarse en disco, puesto que los ficheros de programas no pueden ser leídos de cinta. Si el fallo no fuera encontrado o si se encontrara en la operación algún error de disco, el programa se cancelará.

Después de esto, se te pide que introduzcas el nombre deseado del programa expandido que va a ser creado (líneas 410-420). Este nombre no puede ser el mismo que el que tenía el programa original o que tenga cualquiera de los ficheros que se encuentran actualmente en el disco. Si ya existe cualquier fichero con un nombre igual al que le proporcionas, o si se encuentra que se produce algún tipo de error de disco, el ordenador informará de estos errores y el programa terminará de ejecutarse.

"Expansor" trata el programa que va a ser descompactado como si fuese un fichero secuencial en disco. Según va leyendo el programa original, cada uno de los números de línea se va visualizando en la pantalla (líneas 490-510). Esto puede servirte como indicación de la manera en que van progresando las cosas según se ejecuta el programa "Expansor"; puede resultar un proceso bastante lento.

Después de que el "Expansor" copia el número de la línea original, lleva la línea real a una matriz C (líneas 550—

IEAS

560) y después lee el siguiente enlace y el siguiente número de línea (líneas 600-620). Cuando encuentra el enlace cero al final del programa, el número de línea siguiente se fuerza a que sea 64.000. Este número es mayor que cualquier número posible en una línea de programa Basic, forzando, de esta forma, un manejo adecuado de la última línea del programa que se está leyendo.

Una vez que se ha leído completamente una línea entera y se comprueba que no es más larga que el límite que has impuesto sobre la longitud de la línea, comienza a buscar el signo de dos puntos y determinados caracteres Basic (líneas 660-940), comprobando carácter a carácter. Si la línea es más corta que la longitud especificada con límite, se copia dejándola intacta (línea 860). Si uno de los caracteres especiales Basic es encontrado (líneas 820-850) el resto de la línea se copia de forma intacta.

Cuando se encuentra un signo de dos puntos, se divide la línea siempre que el número de línea actual, mas uno, sea menor que el número de la línea siguiente del programa original (líneas 700-760). La línea actual se escribe en el disco con el enlace apropiado y el flag de terminación. Los dos puntos que se encuentren solos al comienzo del cualquier línea se dejan como están, mientras que los dos puntos, o los espacios, a mitad de una línea se borran (línea 750).

Asegúrate siempre que tecleas el comando completo —PRINT#—

Siempre que los signos de comillas aparezcan en la línea, se copia el resto de la línea dejándola intacta hasta que se cierran las comillas o se encuentre el final de la línea (líneas 910-940). Al final del programa, se escribe un enlace cero en el disco para terminar de forma adecuada el nuevo programa y se cierran todos los ficheros.

Las subrutinas se colocan cerca del principio del programa para ayudar a que vayan las cosas más rápido. La subrutina que comienza en la línea 230 lee dos bytes, mientras que en la 240 se lee un byte únicamente del fichero del programa original. Cuando el programa vuelve de esta subrutina, el último carácter leído se encuentra en C$, con los códigos de carácter en V y en VI. Las líneas 270-290 comprueban los errores de disco y, o bien se vuelve a la línea que había hecho la llamada, o bien se visualiza el error del disco y se deja de ejecutar el programa. Las líneas que van de la 300 a la 320 sirven para calcular el enlace correcto que hay que utilizar en L$ y, asimismo, se utilizan para escribir la línea completa, de acuerdo con el enlace, al fichero del nuevo programa.

Los programas que se crean de esta forma, por tanto descompactados se encuentran completamente enlazados y preparados para ejecutarse. Puedes utilizar el Descompactador II y el Compactador II sobre cualquier programa Basic normalizado que no contenga rutinas en lenguajes ensamblador dentro del programa.

Según se mencionó en el artículo Compactador II, no se debe utilizar un signo de interrogación como abreviatura del Print en los comandos Print #. La línea sigue listándose correctamente, pero de forma interna, el código resultará ser incorrecto y, por consiguiente, generará el correspondiente error de Sintaxis si se fuera a ejecutar. Asegúrate siempre que tecleas el comando completo —PRINT#— para evitar problemas.

```
100 REM ********************************
**
110 REM
120 REM           E X P A N S O R
130 REM
140 REM      POR  ROBERT W. BAKER
150 REM 15 WINDSOR DR., ATCO, NJ 080
04
160 REM
170 REM ********************************
**
180 :
190 GOTO 360
200 :
210 REM >>>>>>> SUBRUTINAS <<<<<<<
220 :
```

```
230 GOSUB 240: V1=V
240 GET#5,C$: GOSUB 270
250 V=0: IF C$<>"" THEN V=ASC(C$)
260 RETURN
270 INPUT#15,EN,EM$,ET,ES: IF EN=0 T
HEN RETURN
280 PRINT: PRINT "ERROR[SPC]EN[SPC]E
L[SPC]DISCO":PRINT
290 PRINT EN;EM$;ET;ES: GOTO 1000
300 IF L$="" THEN RETURN
310 LA=LA+LEN(L$)+2: L1=INT(LA/256):
 L2=LA-(L1*256)
320 PRINT#6,CHR$(L2);CHR$(L1);L$;: L
$="": RETURN
330 :
340 REM ***** INICIALIZACION *****
```

EXPANSOR DE LINEAS

```
350 :
360 PRINT"[CLR][10SPC]E[SPC]X[SPC]P[SPC]
A[SPC]N[SPC]S[SPC]O[SPC]R": PRINT: P
RINT
370 PRINT"LONGITUD[SPC]MINIMA[SPC]DE
[SPC]LAS[SPC]LINEAS": INPUT"[3SPC]1[3CRSRL]
";XL
380 DIM C(256): CLOSE 15: OPEN 15,8,
15
390 PRINT: PRINT"NOMBRE[SPC]DEL[SPC]
PROGRAMA": INPUT F1$
400 OPEN 5,8,5,"0:"+F1$+",P,R": GOSU
B 270
410 PRINT: PRINT"NOMBRE[SPC]DEL[SPC]
PROGRAMA[SPC]DE[SPC]SALIDA": INPUT F
2$
420 OPEN 6,8,6,"0:"+F2$+",P,W": GOSU
B 270
430 PRINT:PRINT"OK,[SPC]PROCESANDO[SPC]
LINEA": PRINT
440 GOSUB 230: PRINT#6,CHR$(V1);C$;
450 LA=V1+(256*V): L$="": GOTO 600
460 :
470 REM ***** SACA LINEA
480 :
490 GOSUB 300: IF LK=0 THEN 980
500 LN=NL: PRINT LN;"[6SPC]": PRINT"
[CRSRU]";
510 L$=CHR$(LL)+CHR$(LH)
520 :
530 REM ***** LEER LINEA DE PROGRAMA
 BASIC
540 :
550 X=1
560 GOSUB 240: C(X)=V: IF V>0 THEN X
=X+1: GOTO 560
570 :
580 REM ***** OBTENER LINK Y NUMERO
DE LINEA
590 :
600 GOSUB 230: LK=V+V1: IF LK=0 THEN
 NL=64000: GOTO 620
610 GOSUB 230: NL=V1+(256*V): LL=V1:
 LH=V
620 IF L$="" THEN 490
630 :
640 REM ***** CORTAR LA LINEA SI ES
POSIBLE
650 :
660 V=X: X=1: IF V<XL THEN 860
670 :
680 REM SALTAR SI NO SON DOS PUNTOS
690 :
700 IF C(X)<>58 THEN 820
710 IF X=1 THEN L$=L$+CHR$(C(X)): GO
TO 750
720 LN=LN+1: IF LN>=NL THEN 860
730 H=INT(LN/256): L=LN-(256*H)
740 L$=L$+CHR$(0): GOSUB 300: L$=CHR
$(L)+CHR$(H)
750 X=X+1: IF C(X)=32 OR C(X)=58 THE
N 750
760 GOTO 700
770 :
780 REM COPIAR EL RESTO DE LA LINEA
SI HAY:
790 REM  GOTO, END, RUN, IF, RETURN
800 REM  REM, STOP, LIST, CONT
810 :
820 IF C(X)<128 OR C(X)>155 THEN 910
830 IF C(X)=128 OR C(X)>153 THEN 860
840 IF C(X)<137 OR C(X)>144 THEN 930
850 IF C(X)=140 OR C(X)=141 THEN 930
860 L$=L$+CHR$(C(X)): IF C(X)>0 THEN
 X=X+1: GOTO 860
870 GOTO 490
880 :
890 REM SALTA SI NO SON COMILLAS
900 :
910 IF C(X)<>34 THEN 930
920 L$=L$+CHR$(C(X)): X=X+1: IF C(X)
<>34 AND C(X)<>0 THEN 920
930 L$=L$+CHR$(C(X)): IF C(X)>0 THEN
 X=X+1: GOTO 700
940 GOTO 490
950 :
960 REM *** FIN DEL PROGRAMA BASIC
970 :
980 PRINT#6,CHR$(0);CHR$(0);
990 PRINT: PRINT: PRINT"HECHO":PRINT
:PRINT
1000 CLOSE 5: CLOSE 6: CLOSE 15
```

Por Guy WRIGHT • Adaptado por Alvaro IBAÑEZ

Tus primeros pasos en BASIC

Si quieres comunicarte con tu ordenador, tendrás que aprender a hablar su mismo lenguaje. Este artículo explica algunos conceptos de Basic y frases que te situarán en el buen camino para que puedas llegar a programar.

Intentar aprender Basic es como intentar aprender una lengua extranjera. Tienes que memorizar algunas de las palabras y aprender cómo se ponen juntas en frases que tengan algún significado.

Afortunadamente, sólo hay alrededor de 63 palabras en Basic y únicamente un puñado de "reglas gramaticales" que tengas que aprender. La parte más dura de aprender es la lógica de la programación y cómo detectar o depurar los problemas del programa.

Sirvientes y tostadores

Los ordenadores son más o menos tan listos como un tostador. Tienes que ir diciendo al ordenador exactamente qué es lo que tiene que hacer a través de un programa, que es un conjunto, paso a paso, de instrucciones escritas en inglés computerizado que se denomina BASIC. Cada una de las instrucciones debe ser suficientemente simple para que pueda entenderla el ordenador, lo cual muchas veces no es nada sencillo de conseguir.

Imaginate que un sirviente muy rápido, pero muy tonto al mismo tiempo, va a ir a tu casa a hacer algo que tú quieres que haga. El problema estriba en que tú no estarás en casa cuando venga para poder verle, luego tienes que dejarle todas tus instrucciones escritas en tarjetas numeradas.

Cada tarjeta puede contener únicamente una instrucción y el sirviente debe seguir las tarjetas en el orden que las has marcado. Si quieres que él barra el vestíbulo, en primer lugar tienes que darle instrucciones de que tome una escoba del armario, o por el contrario, se sentirá perdido y sin saber qué hacer cuando llegue a la instrucción "barrer".



Puedes, de esta forma, empezar a vislumbrar hasta qué punto podrían ponerse las cosas difíciles si quisieras que el sirviente hiciese el balance de tus talonarios o mantuviese un seguimiento de las finanzas de tu negocio.

Un peligro que corren todos los nuevos programadores es intentar atacar un problema que es demasiado complicado. Tienes que aprender una serie de conceptos, a veces algo dificultosos, cuando comiences.

Una razón por la cual resultan a veces un tanto complicados de entender estos conceptos es que los libros que tratan sobre temas de programación generalmente no están escritos en el estilo de lenguaje que estamos acostumbrados a manejar la mayoría de nosotros en nuestras conversaciones. Este artículo utiliza algunos términos que pueden sonar un poco tontos, pero también pueden resultar más claros y más fáciles de entender.

Variables

Piensa en las variables como si fuesen los bolsillos del mono de un carpintero.

Cada bolsillo lleva una etiqueta que es, o bien una letra y un número, o bien dos letras (por ejemplo, LH para el bolsillo de la cadera izquierda o H1 y H2 para el bolsillo 1 y para el bolsillo 2). Además, cada bolsillo puede contener únicamente cierto tipo de cosas.

Hay un cierto número de diferentes clases de bolsillos de ordenador, pero únicamente tenemos que preocuparnos por ahora por dos tipos determinados: bolsillos que pueden contener solamente números y aquellos que pueden contener cualquier cosa que teclees en el teclado (letras, números, teclas de cursor, etc.).

En la jerga informática, a estas dos clases de bolsillos se les denomina variables numéricas y variables de cadena (string). Una manera sencilla de diferenciarlas es mediante un signo de dólar que se pone al final de la etiqueta de cualquier variable de cadena. AA es la etiqueta de una variable numérica y AA$ es la etiqueta que corresponde a una variable de cadena que puede contener cualquier cosa. Probablemente sería más sencillo si únicamente hubiese una sola clase de bolsillo, pero los ordenadores necesitan que se les indique cuándo tienen que sumar dos números o cuándo tienen que sumar dos letras.

Puedes pensar que los bolsillos con $ son los que pueden contener palabras (o cadenas de caracteres). Algunas veces las palabras tienen sentido ("perro", "gato", "uno"); sin embargo, otras veces no tienen porqué tenerlo (CO87JN).

Si le dices al ordenador que sume las palabras "gato" y "perro", te dará como resultado "gatoperro". Si, por el contrario, le dices que sume 1 y 2 dará como resultado 3. Pero si le dices que sume las "palabras" "2" y "4" te proporcionará un resultado de 24. (Recuerda, una palabra puede ser cualquier carácter o combinación de caracteres que se tecleen en el teclado).

También tienes que aprender cómo el ordenador mueve la información hacia dentro y hacia afuera de los bolsillos y cómo lleva a cabo operaciones aritméticas sobre cosas que ya se encuentran en los bolsillos. Un signo igual (=) significa "introduce en este bolsillo todo lo que haya a la derecha del signo = ". De esta forma, si quieres que el ordenador introduzca un 6 en el bolsillo LH, tienes que teclear LH = 6. Si quieres poner la palabra GATO dentro del bolsillo denominado CA$, tendrás que teclear CA$ = "CAT". Las comillas son esenciales y obligatorias en las variables de cadena.

Si quieres que el ordenador sitúe el resultado de la suma de dos números en un determinado bolsillo, tendrás que teclear RE = 7 + 4. El ordenador, en primer lugar, va a sumar los números y, después, va a colocar la respuesta a dicha operación en el bolsillo cuya etiqueta sea RE.

Si quieres que el ordenador tome un número de un bolsillo, le sume al número que se encuentre en otro bolsillo y, posteriormente, introduzca el resultado de dicha suma en un tercer bolsillo, tienes que teclear TH = FJ + SE. En este caso, el ordenador tomaría el número del bolsillo cuya etiqueta fuese FI (el primero), le sumaría al número que se encontrara en el bolsillo denominado SE (el segundo) y pondría el resultado de dicha suma en el bolsillo cuya etiqueta fuese TH (el tercero). Recuerda que todo lo que se encuentre a la derecha del signo de igualdad se realiza en primer lugar y después es cuando se introduce el resultado en el bolsillo que está indicado en la parte izquierda del signo igual.

Tomando todo esto en cuenta, tiene, por tanto, sentido el teclear AN = AN + 6, por ejemplo. Si la línea fuese AN = 4 + 6, el ordenador, en primer lugar, tendría que sumar 4 y 6 y, posteriormente, tendría que almacenar la respuesta en el bolsillo que va a contener dicho resultado y que se ha denominado como AN. Si la línea fuera AN = FI + 6, el ordenador, en primer lugar, tendría que buscar el número que se encuentra almacenado en el bolsillo FI, sumarle a 6 y, después, almacenar el resultado en AN. Por lo tanto, si tecleas AN = AN + 6 y pulsas la tecla RETURN, en primer lugar, el ordenador va a buscar el número que ya se encuentra en AN, le suma a 6 y, después, introduce el nuevo número resultante otra vez en la variable AN.

Vale la pena insistir un poco hasta que entiendas el concepto de las variables. Prueba unos cuantos experimentos en tu ordenador y mira a ver qué es lo que sucede con cada uno de ellos. ¡No hay nada que puedas teclear en el teclado del ordenador que pueda dañar al ordenador!

Operaciones de ordenador

Todas las diferentes acciones que puede llevar a cabo un ordenador, pueden dividirse en tres tipos: conseguir información (entrada), hacer alguna cosa con dicha información (proceso), después poner esa información en algún sitio (salida).

Cada tarjeta puede contener únicamente una instrucción y el sirviente debe seguir las tarjetas en el orden que las has marcado. Si quieres que él barra el vestíbulo, en primer lugar tienes que darle instrucciones de que tome una escoba del armario, o por el contrario, se sentirá perdido y sin saber qué hacer cuando llegue a la instrucción "barrer".

Entrada

La entrada representa el proceso por el que se introduce algún tipo de información dentro del ordenador, de tal forma que pueda realizarse algo con ella. Uno de los bloques de información más importantes con que debe ser alimentado el ordenador es el programa en sí mismo (de cualquier otra manera, el ordenador no podrá saber qué es lo que tiene que hacer cuando le llegue el resto de la información).

El primer paso es teclear el programa. Puedes hacer dos cosas, o bien puedes ejecutar el programa después de haberlo tecleado o bien puedes salvarlo para que sea cargado de nuevo en el ordenador más tarde. En cualquier caso, el programa debe introducirse de alguna forma en el ordenador.

P.V.P. 2.950 Ptas.

Precio adjuntando boletín: **2.500 Pts.**

REGLA RELOJ CALCULADORA

- ***Reloj con posición "vertical" para más fácil lectura.***
- ***Calculadora "extraíble" de la regla con las 4 funciones elementales (+, —, ×, :), posee raíz y % así como memoria.***
- ***Tabla de conversión de medidas impresa en la regla.***

GRAN PINZA

(en madera barnizada) Sirve tanto de pisapapeles como para sujetarlos, lleva impresos los distintivos de todas nuestras publicaciones.



P.V.P. 400 Ptas.

Precio adjuntando boletín: **300 Ptas.**



Si eres "commodoriano" ... ¡¡¡Que lo sepan!!!



CAMISETA DE FELPA

- ***Estampada en el anverso con el distintivo de Commodore World y en el reverso con el distintivo de las publicaciones hermanas, MicroSistemas y PC World.***

P.V.P. 1.950 Ptas.

Precio adjuntando boletín: **1.500 Ptas.**

BOUTIQUE Commodore World - Boletín de Pedido

NOMBRE ....
DIRECCION ....
POBLACION ....
(C.P. ....) PROVINCIA ....
TELEF.: ....

FORMA DE PAGO

☐ Cheque por valor de .... pesetas
☐ Envío giro nº ...... por .... pesetas
+ 100 Ptas. por gastos de envío por unidad.

DESEO ME ENVIEN LA CANTIDAD DE:

☐ REGLAS RELOJ-CALCULADORAS A 2.500 Ptas. c/u.
☐ CAMISETAS DE FELPA A 1.500 Ptas. c/u.
TALLAS MEDIANAS ☐ o GRANDE ☐
☐ GRAN PINZA A 300 Ptas. c/u.

ENVIAR A COMMODORE WORLD • C/BARQUILLO, 21-3º IZQDA. • 28004 MADRID



COMERCIAL MORON

ERCILLA, 12 - TELEFONO 468 26 93
28005-MADRID
Para información telefónica solamente de 4-5 tarde
Preguntar por el Sr. Morón

Deseo recibir información completa del QUICK DATA DRIVE y de la Caja de Ritmos sin ningún compromiso por mi parte:

Nombre ....

Dirección ....

Télef.: .... Ciudad: ....
C.P.: .... Provincia: ....

Soy distribuidor ☐   Soy particular ☐

(*) Marcar con un asterisco lo que interese.



La manera más sencilla es comprar el programa a alguna otra persona y, entonces, cargarlo en el ordenador con un Datasette o con una unidad de disco. Pero si quieres escribir tus propios programas, tienes que disponer de algún método para salvarlos. De cualquier otra forma, en cada ocasión que quieras utilizar el programa, tendrás que teclearlo completo desde el principio hasta el final una y otra vez. Esto puede estar bien si es un programa muy corto o si es alguno que vayas a utilizar solamente una vez.

Sin embargo, generalmente es mejor escribir un programa, salvarlo en cinta o en disco, depurarlo, modificarlo o hacer lo que se quiera con él, sabiendo que si se destruyera en algún momento, tienes una copia de backup.

Un detalle a tener en cuenta: cuando salves un programa en cinta o en disco, todo lo que hubiera en cualquiera de los bolsillos (variables) se desprecia, por lo que si quieres que el programa mantenga ciertas cosas en ciertos bolsillos, tendrás que escribir al principio dichos datos en el programa. Por ejemplo, si quieres que aparezcan las palabras "Hola, Sam" cuando se ejecute el programa, tendrás que introducir dichas palabras en el programa.

Tan pronto como se haya introducido el programa en el ordenador, hay otra clase de información que tiene que ser añadida mientras que esté ejecutando el programa (incluso en el caso de que seas la única persona que jamás vaya a utilizar el programa, va a haber algunas veces que tú mismo quieras introducir nueva información.

Si quieres diseñar un programa que sume los números suministrados por el usuario, debes conseguir que tales números se puedan introducir en el programa mientras éste se está ejecutando. Cualquier tipo de información que se introduzca en el ordenador mientras se esté ejecutando el programa, puede considerarse que es una entrada (por ejemplo, puede introducirse un nombre, pueden introducirse números o incluso el movimiento de un Joystick). El comando que se utiliza con una mayor frecuencia para introducir información en el ordenador mientras se está ejecutando un programa es la instrucción Input.

Cuando el ordenador detecta la palabra Input, cesa la ejecución del programa y espera a que se teclee la correspondiente información y que sea pulsada la tecla Return. Entonces, el ordenador coloca lo que se haya introducido en el bolsillo que se hubiera especificado y, a continuación, prosigue la ejecución del programa.

Esto significa que la variable que hayas señalado en el comando Input debe consistir en, o bien un bolsillo numérico (si esperas que se vaya a introducir un número) o un bolsillo de cadena de caracteres (si esperas que pueda introducirse cualquier cosa). Si la respuesta que se haya pulsado a la petición de entrada, no corresponde con el tipo de bolsillo, obtendrás un mensaje de error.

Proceso

La siguiente actividad que puede desarrollar un ordenador se denomina proceso. Esto consiste en la operación que el ordenador lleva a cabo con la información que se haya introducido en el ordenador como cualquiera de las entradas que se han descrito anteriormente.

Muchas áreas de proceso tienen sus propios nombres y pueden diferenciarse según un conjunto de categorías.

Bucles, o acciones repetitivas, que indican al ordenador que lleve a cabo alguna cosa una y otra vez durante un número determinado de veces.

Operaciones aritméticas como son la suma, la resta, la multiplicación, etc., que los ordenadores llevan a cabo con una infalibilidad y una rapidez cegadora.

Subrutinas que son mini-programas que hay dentro de un programa y que pueden utilizarse una y otra vez.

Toma de decisiones (en un lenguaje coloquial) que se llevan a cabo por parte de la gran mayoría de los programas en determinados momentos, incluso aunque lo único que se tenga que decidir no sea otra cosa que cuando se finaliza la ejecución del programa.

Almacenamiento y recuperación de datos que incluye el traslado de información de un sitio a otro.

Bucles

Hay dos formas básicas de conseguir que el ordenador haga alguna cosa una y otra vez de forma repetitiva. Para volver a nuestra analogía anterior, la forma más sencilla es decirle que vaya de vuelta a una línea determinada o a un número de "tarjeta" en particular y comience a seguir las intrucciones desde allí. El sirviente lee las tarjetas 1, 2, 3, 4 y 5, y cuando lee la tarjeta número 6, ésta dice algo parecido a "vuelve a la tarjeta número 2 y comienza a hacer de nuevo todo desde allí". Esto conduce, sin embargo, a un nuevo problema.

Siendo como es literalmente un bruto mental, el ordenador llevará a cabo las tareas de cada una de las tarjetas (si se encuentran escritas con las reglas correctas), llegará a la tarjeta 6, comenzará de nuevo desde la 2, llegará otra vez hasta la 6, de vuelta a la 2, de nuevo hasta la 6, entonces vuelve a la 2, y continuará de esta manera, ad infinitum. Seguirá haciendo las mismas cosas una y otra vez hasta que tires del enchufe o pulses la tecla de parada.

Esto es lo que se denomina un bucle sin fin, por razones obvias. Hay diferentes formas de solucionarlo.

Si el paso 3 dijera "suma 1 al número que se encuentra en el bolsillo LH", y después el paso 4 dijese "si el número que se encuentra en el bolsillo LH es igual a 47, entonces finaliza el programa", entonces el ordenador repetiría los pasos, que van desde el 2 hasta el 6, un total de 47 veces. Cada vez que se pasara a través del bucle, se añadiría 1 a LH y cuando LH fuese igual a 47, el programa terminaría.

(Aún hay un programa potencial aquí. Si el número que se encuentra en el bolsillo LH fuera, por alguna razón, en algún momento igual a 50, entonces LH se iría haciendo cada vez más grande, pero en ningún momento llegaría a ser igual a 47. De este modo, el ordenador habría entrado de nuevo en un bucle sin fin).

La idea de situar un "test" dentro de un bucle se denomina flag. En Basic, ya hay un tipo determinado de bucles, llamados bucles FOR... NEXT, que ya vienen provistos de un flag. Tienes que indicar al ordenador cuantas veces quieras que se repita la operación y cuando se alcance dicho número, el ordenador continúa con la siguiente instrucción del programa.

El formato de un bucle For... Next es como se indica a continuación:

FOR bolsillo = número de comienzo TO número de finalización.

y, posteriormente, en el programa tiene que incluirse la sentencia:

NEXT bolsillo

En nuestro ejemplo, la tarjeta número 2 podría leerse como FOR LH = 1 TO 47, y la tarjeta número 6 tendría que decir NEXT LH. Cada vez que el ordenador llegara a la tarjeta número 6, saltaría a la tarjeta número 2 hasta que LH fuese igual a 47. Cuando sucediera tal cosa, el ordenador saltaría a la siguiente tarjeta que sería la número 7.

Si quieres que el ordenador imprima la palabra Hola doce veces y después se pare, el programa podría ser algo como lo que se muestra a continuación:

```
10 FOR RH = 1 TO 12
20 PRINT "HOLA"
30 NEXT RH
40 END
```

RH es el bolsillo flag que seguirá la pista del número de veces que el ordenador ha recorrido el bucle. Generalmente, también, numeramos las tarjetas (líneas del programa) de diez en diez (el ordenador no tiene en cuenta cuáles son los números que hay en cada línea; siempre va ejecutando en secuencia desde los números más bajos a los números más altos, a menos que le indiquemos que lo haga de otra manera). Asimismo, la numeración por decenas deja huecos libres para poder insertar líneas de programa con posterioridad, si así se necesitase.

En el ejemplo del programa anterior, la línea número 10 establece el bucle. Aquí, indicamos al ordenador qué variable es la que tiene que utilizar como flag (RH), con qué número queremos que empiece la cuenta (1 en el ejemplo) y hasta qué número queremos que llegue la cuenta (12 en el ejemplo) antes de que finalice el bucle y se continúe con el programa.

La línea 20 imprime nuestro mensaje en la pantalla (salida).

La línea 30 indica al ordenador que vaya de nuevo a la línea donde comienza el bucle (línea 10) y comience de nuevo otra vez.

La línea 40 indica al ordenador que se finalice con la ejecución del programa. Esta es la línea a la cual irá el ordenador cuando se haya repetido el bucle dos veces, como se le había indicado, ya que siempre que se termina de realizar un bucle completo, se salta, para continuar con el programa, a la línea siguiente a la línea que contiene el NEXT del bucle FOR... NEXT que acaba de finalizarse.

Operaciones aritméticas

Las operaciones aritméticas son la clase de operaciones que mejor llevan a cabo los ordenadores. Estos suman, restan y multiplican números con una velocidad increíble. Teclea en tu ordenador:

```
PRINT 44459. 2576 / 456. 32
```

y antes de que puedas tomar un respiro, imprimirá 97.43.

Si quieres que un programa únicamente lleve a cabo una manipulación rápida y correcta de ciertos números, entonces te has dirigido al lugar idóneo. El truco consiste en desarrollar la fórmula, de manera que el ordenador pueda ejecutar todo en el orden correcto. Si tienes una fórmula larga de gran complejidad que te gustaría que el ordenador te la calculara, hay unas cuantas reglas sencillas que puedes seguir. O bien divides la fórmula en pasos más pequeños, poniendo cada uno de los pasos en líneas independientes, o bien sitúas las operaciones, que quieras que se lleven a cabo en primer lugar, encerradas entre paréntesis, por ejemplo, introduce:

```
PRINT 12 * (5 + 32) — (43.65 + 78)
```

o bien utiliza este programa:

```
10 x = 5 + 32
20 y = x * 12
30 z = 43.65 + 78
40 LH = Y — Z
50 PRINT LH
```

Esto último es más sencillo de entender y, también, es menos probable que te puedas confundir al introducirlo en el ordenador. (También es un buen ejemplo de movimiento de la información entre los diferentes bolsillos).

Subrutinas

Las subrutinas son miniprogramas que se utilizan para llevar a cabo una serie de instrucciones muchas veces dentro de un programa. Un ejemplo es una rutina que imprima un menú de opciones diferentes en la pantalla. Una vez que selecciones una opción, el ordenador sale del menú para llevar a cabo dicha opción, volviendo después de nuevo a imprimir el menú otra vez.

Para enviar al ordenador a la subrutina, teclea GOSUB y el número de la línea donde comienza la subrutina. El ordenador salta a dicha línea, lleva a cabo las tareas que se encuentren especificadas en la subrutina y, después, cuando encuentra la palabra de Basic RETURN que se sitúa después de la última instrucción, el ordenador salta de vuelta a la línea de programa situada inmediatamente después de la instrucción GOSUB. Una subrutina de menú sencilla podría ser la siguiente:

```
10 GOSUB 100
20 IF LH = 1 THEN PRINT"ESTO"
30 IF LH = 2 THEN
   PRINT"AQUELLO"
40 IF LH = 3 THEN PRINT"OTRA
   COSA"
50 GOTO 10
100 PRINT"ELIGE UN NUMERO"
110 PRINT"ENTRE 1 y 3"
120 PRINT
130 PRINT"1 — PRINT ESTO"
140 PRINT"2 — PRINT
    AQUELLO"
150 PRINT"3 — PRINT OTRA
    COSA"
```

El programa anterior comienza dirigiéndose a la subrutina que empieza a partir de la línea 100. Las líneas 100-150 imprimen el menú en la pantalla (la línea 120 sólo imprime una línea en blanco para hacer que la pantalla aparezca más clara. La línea 160 indica al ordenador que permanezca en situación de espera hasta que el usuario teclee alguna cosa y pulse la tecla Return. Tan pronto como se pulsa la tecla Return, el número que se teclea se lleva a ocupar el bolsillo (variable) denominado LH. La línea 170 envía al ordenador de nuevo de vuelta a la línea 10. Después de esto, en las líneas 20 a la 40, el ordenador comprueba el número situado en LH para verificar si es igual a 1, 2 ó 3. Si LH corresponde a uno de estos números, entonces se imprime el mensaje adecuado en la pantalla y la línea 50 envía al ordenador de vuelta para que comience otra vez el procedimiento completo, es, por tanto, un bucle sin fin.

Es esencial que el ordenador no se encuentre con la palabra RETURN en un programa a menos que ya haya sido enviado a una subrutina con la palabra GOSUB. Si borrases la línea número 50 y dejases que el programa se ejecutara, el ordenador iría a la subrutina (líneas 100-170), imprimiría el menú antedicho, seleccionaría una de las opciones, volvería a la sección de imprimir el menú de nuevo, esperaría una nueva selección y después, cuando llegase al RETURN, no sabría dónde volver y daría un mensaje de error que dijera RETURN sin GOSUB.

La mayor parte de los programadores ponen todas las subrutinas de un programa dentro de una porción de los programas y dirigen al ordenador hasta este área mediante sentencias GOTO. (GOTO simplemente indica al ordenador que salte hacia delante o hacia atrás hasta una línea que, generalmente, no se encuentra en secüencia con el resto de las líneas que viene ejecutando el programa y para continuar después desde dicha línea).

Si le dices al ordenador que sume las palabras "gato" y "perro", te dará como resultado "gatoperro". Si, por el contrario, le dices que sume 1 y 2 dará como resultado 3. Pero si le dices que sume las "palabras" "2" y "4" te proporcionará un resultado de 24. (Recuerda, una palabra puede ser cualquier carácter o combinación de caracteres que se tecleen en el teclado).



Toma de decisiones

La toma de decisiones, para un ordenador, tiene que ver básicamente, en un ordenador, con la comparación de cosas; por lo tanto, dependiendo de los resultados de dicha comparación, el ordenador irá a una o a otra parte del programa. En el ejemplo expuesto en último lugar, se utilizó la forma más común de toma de decisiones, la sentencia If... Then.

Esto significa "Si la comparación siguiente resulta verdadera, entonces continúa realizando todo lo que venga detrás de la palabra THEN; si la comparación da como resultado que es falsa, entonces continúa con la línea siguiente del programa". Una comparación puede realizarse con =, , , = o = (igual a, menor que, mayor que, no igual a, igual a o mayor que, igual a o menor que, respectivamente).

Por lo tanto, en el ejemplo, la línea número 20 comprueba si LH es igual a 1. Si el resultado de la comparación es positivo, entonces el ordenador imprime la palabra ESTO. Si LH resulta que no es igual a 1, entonces el ordenador automáticamente se dirige a la siguiente línea, para continuar con el programa, que es la lína 30, donde tiene lugar otro "test". Si el menú, como respuesta a éste, el usuario ha introducido un 5, en lugar de 1, 2 ó 3, entonces el ordenador llevará a cabo los tres "test" que se encuentran seguidos, volverá de nuevo a la línea 10 y comenzará otra vez el programa completo, sin imprimir ningún mensaje en este caso.

Almacenamiento y recuperación

El almacenamiento y la recuperación son funciones que requieren una cierta habilidad en programación compleja. Si quieres que un programa te permita seguir la pista, además de facilitársela, de todas las modificaciones que se realicen con la información, utiliza una que ya haya sido escrita hasta entender la programación, las cintas de ficheros, las entradas/salidas de disco, etc. Cuando consigas esto, podrás escribir tus propios programas o modificar otros.

Puedes escribir programas que contengan mucha información, pero puede resultar difícil modificar dicha información. Si únicamente quieres que el ordenador mantenga una lista, entonces teclea la información con los números de línea delante de cada uno de los componentes de la lista. De esta manera, cuando quieras ir a algún punto de la lista, tendrás que cargar el programa y, en lugar de teclear RUN, teclea LIST para ver el programa. Si quieres cambiar alguna parte de la información contenida, entonces carga el programa, modifica las líneas que quieras cambiar y vuelve a salvarlo.

Si lo que quieres es que tu ordenador imprima líneas individuales sin que tengas que utilizar una rutina de búsqueda, puede hacerlo comenzando cada línea con un comando Print. Por ejemplo:

```
10 PRINT "TIA SARA 123 - 456 -
7890"
20 PRINT "TIO JESSIE 890 - 567 -
1234"
```

Probablemente éste sea uno de los programas más rudimentarios que pueden utilizarse para mantener un listín telefónico, pero funcionará y es muy fácil de escribir.

Salida

El aspecto final de cualquier programa es obtener información del ordenador mediante una salida a la pantalla o a la impresora. Puedes indicar al ordenador que sume los números (A = 2 + 3) y lo llevará a cabo a una velocidad de luz y te devolverá un READY y el cursor intermitente que señala que ha finalizado la suma. Sin embargo, si no le indicas que imprima la respuesta (PRINT A) jamás podrás ver el resultado de dicha operación.

El comando Print es, con toda probabilidad, el comando que se utiliza con más frecuencia en Basic; sin él, no tenemos ninguna forma de saber qué es lo que está ocurriendo dentro del ordenador.

Una sentencia Print puede tener varios formatos. PRINT "xxx" imprimirá en la pantalla todo lo que pongas dentro de las comillas (incluso movimientos de cursor o determinados símbolos como SHIFT CLR). Si tecleas PRINT, y entonces mantienes pulsada la tecla de mayúsculas al mismo tiempo que pulsas la tecla CLR / HOME, aparecerá un corazón invertido. Añade el signo de cerrar comillas, y pulsa la tecla Return. Si has llevado a cabo todo lo anterior en el orden correcto, se borrará la pantalla.

Otro formato del comando Print es PRINT xx, donde xx es el nombre de una variable. Esta instrucción imprimirá el contenido del bolsillo denominado xx en la pantalla. También puedes utilizar el ordenador como si fuera una calculadora. Por ejemplo, PRINT 3 + 4 imprimirá 7 en la pantalla.

Mantén la moral

Todos los problemas de programación pueden dividirse en trozos más pequeñas y dichos trozos pueden romperse, a su vez, en trozos más pequeños, hasta que puedas hecer corresponder los comandos que tú conoces con los diferentes trozos. Según se van haciendo las partes más y más pequeñas, encontrarás que cada vez un número mayor de ellas contienen palabras Basic que llevan a cabo justamente la cosa que se pretende hacer con esa parte.

Aprender a programar es una experiencia increíblemente frustrante para todo el mundo. El que, en un principio, puedas encontrar más mensajes de error que respuestas, no significa que seas tonto. No hay programador vivo que no haya pasado el mismo trago.

Sigue leyendo y releyendo los manuales, toma un comando en un determinado momento e imagínate cómo trabaja, mira los programas que puedas sacar de las revistas y pide ayuda a los amigos cuando sea necesario. Si falla todo esto, hay miles de programas que puedes comprar o teclear a partir de revistas. La mayoría de ellos, suponen un desafío. ■

Cuando el ordenador detecta la palabra Input, cesa la ejecución del programa y espera a que se teclee la correspondiente información y que sea pulsada la tecla Return. Entonces, el ordenador coloca lo que se haya introducido en el bolsillo que se hubiera especificado y, a continuación, prosigue la ejecución del programa.



Comandos del MONITOR DE CODIGO/MAQUINA del COMMODORE 16

El MONITOR DE CODIGO/MAQUINA es un programa incorporado en el COMMODORE 16 que permite escribir con facilidad programas en código/máquina. En este utilitario se incluye un mini-ensamblador y un desensamblador.

Introducción

Los programas en código/máquina preparados mediante el MONITOR de C/M pueden funcionar por sí mismos, o ser utilizados como "subrutinas rápidas" en los programas BASIC, ya que el MONITOR de C/M tiene la capacidad de coexistir pacíficamente con el BASIC.

A continuación se da una lista resumida de los comandos del MONITOR de C/M.

A ASSEMBLE: Ensambla una línea de código del 6502.

C COMPARE: Compara dos secciones de memoria e informa de las diferencias.

D DISASSEMBLE: Desensambla una línea de código del 6502.

F FILL: Rellena la memoria con el patrón especificado.

G GO: Inicia la ejecución en la dirección especificada.

H HUNT: Rastrea el contenido de ciertos bytes en la memoria.

L LOAD: Carga un fichero desde cinta o disco.

M MEMORY: Exhibe el contenido de la memoria.

R REGISTERS: Exhibe el contenido de los registros del 6502.

S SAVE: Salva a cinta o disco.

T TRANSFER: Transfiere un código desde una a otra sección de la memoria.

V VERIFY: Compara la memoria con una cinta o disco.

X EXIT: Sale del MONITOR de C/M.

. (punto): Ensambla una línea de código del 6502.

> (mayor que): Modifica la memoria.

; (punto y coma): Modifica el contenido de los registros del 6502.

Utilización del MONITOR de C/M

Introduzca el MONITOR de C/M escribiendo: MONITOR.

El MONITOR de C/M responde exhibiendo los registros del 6502 y haciendo parpadear el cursor. El cursor es el indicador que le hace saber que el MONITOR de C/M está esperando sus comandos.

Descripción de los comandos

COMANDO: A (ENSAMBLA).

PROPOSITO: Introducir una línea de código en ensamblador.

SINTAXIS: A <dirección> <código de operación> <operando>.

<Dirección> es un número hexadecimal que indica la posición de memoria en donde se debe colocar el código de operación.

<Código de operación> es un mnemónico del lenguaje ensamblador standard de MOS TECHNOLOGY, por ejemplo LDA, STX, ROR, etc.

<Operando> cuando se requiere, puede ser uno de los modos de direccionamiento autorizados. (En los modos de página cero, se requiere un número hexadecimal de dos dígitos cuyo valor debe ser inferior a $100. En los restantes direccionamientos, se requiere un número hexadecimal de 4 dígitos).

Se utiliza la tecla RETURN para indicar el final de la línea de ensamblado. Si hay algún error en ella, aparecerá un interrogante indicándolo y el cursor se moverá a la línea siguiente. Se puede utilizar el editor de pantalla para corregir los errores en una línea.

Tras el ensamblado correcto de una línea de código, el ensamblador imprime un mensaje conteniendo la siguiente posición de memoria autorizada para contener una instrucción, por lo cual A y la dirección no deben teclearse más que una vez cuando se escriban programas en código/máquina en el C-16. Ejemplo:

```
.A 1200 LDX #$00
.A 1202
```

NOTA: Un punto (.) es equivalente al comando ASSEMBLE. Ejemplo:

```
.2000 LDA #$23
```

COMANDO: C (COMPARA).

PROPOSITO: Compara dos áreas de memoria.

SINTAXIS: C <dirección 1> <dirección 2> <dirección 3>.

<dirección 1> es un número hexadecimal que indica la dirección inicial del área de memoria a comparar.

<dirección 2> es un número hexadecimal que indica la dirección final del área de memoria a comparar.

<dirección 3> es un número hexadecimal que indica la dirección de inicio de la otra área de memoria con la que se hace la comparación.

Si las dos áreas de memoria son la misma, el MONITOR de C/M imprime un RETURN, indicando que la segunda área de memoria es la misma que la primera. Se imprimen en pantalla las direcciones de todos los bytes de las dos áreas que no se corresponden.

COMANDO: D (DESENSAMBLA).

PROPOSITO: Desensambla el código máquina en mnemónicos y operandos.

SINTAXIS: D[<dirección>] [<dirección 2>]

<dirección> es un número hexadecimal indicando la dirección en donde debe empezar el desensamblado.

<dirección 2> es una dirección en hexadecimal opcional que indica el final del desensamblado.

El formato del desensamblado es ligeramente distinto al formato del ensamblado. La diferencia consiste en que el primer carácter del desensamblado es un punto en vez de una A (para mayor facilidad de lectura) y se lista también el valor hexadecimal del código.

Un listado de desensamblado puede modificarse mediante el editor de pantalla. Haga algunos cambios en los mnemónicos o en los operandos de la pantalla y, a continuación, pulse RETURN. De este modo se introduce la línea y se llama al ensamblador para nuevas modificaciones.

Un desensamblado puede estar paginado. Si escribe una D hará que sean desensamblados en la pantalla los siguientes 20 bytes de código. Ejemplo:

```
D 3000 3004
. 3000 A9 00     LDA #$00
. 3002 FF        ???
. 3003 D0 2B     BNE $3030
```

COMANDO: F (LLENAR)

PROPOSITO: Rellenar una serie de posiciones con un byte especificado.

SINTAXIS: F <dirección 1> <dirección 2> <byte>.

<dirección 1> es la primera posición a rellenar con <byte>.

<dirección 2> es la última posición a rellenar con <byte>.

<byte> es un número hexadecimal de uno o dos dígitos.

Este comando es útil para inicializar estructuras de datos o cualquier otra área RAM. Ejemplo:

F 0400 0518 EA

Rellena las posiciones de memoria $0400-$0518 con $EA (instrucción NOP.)

COMANDO: G (EJECUTAR)

PROPOSITO: Inicia la ejecución de un programa en la dirección específicada.

SINTAXIS: G[<dirección>].

<dirección> es un argumento opcional que especifica el nuevo valor del contador de programa y la dirección en donde se empezará la ejecución. Cuando se omite, dicha ejecución empieza en el PC actual. (El PC actual puede ser visualizado mediante el comando R.)

El comando GO restaura todos los registros (visualizables mediante el comando R) e inicia la ejecución en la dirección de inicio especificada. Se recomienda precaución al utilizarlo. Para volver al MONITOR de C/M tras ejecutar un programa en código/máquina, utilice la instrucción BRK. Ejemplo:

G 140C

La ejecución empieza en la posición $140C.

COMANDO: H (CAZAR).

PROPOSITO: Rastrear el contenido de una serie de bytes a través de la memoria, dentro de un rango especificado.

SINTAXIS: H<dirección 1> <dirección 2> <datos>.

<dirección 1> es la dirección de inicio del procedimiento de rastreo.

<dirección 2> es la dirección final del procedimiento de rastreo.

<datos> es una serie de datos a buscar. Pueden estar en hexadecimal o en forma de cadena ASCII. ASCII se especifica procediendo el primer carácter por una comilla, por ejemplo 'CADENA. Los datos pueden ser un argumento simple o múltiple. Si es múltiple y en hexadecimal, cada número debe ir separado por un espacio. Ejemplo:

H C000 FFFF 'READ
Busca una cadena ASCII
leyendo desde C000 a FFFF.

H A000 A101 A9 FF 4C
Busca los datos $A9, $FF,
$4C, desde A100 hasta A101.

COMANDO: L (CARGA).

PROPOSITO: Carga un fichero desde cassette o disco.

SINTAXIS: L <"nombre fichero">, <dispositivo>.

<nombre fichero> es cualquier nombre de fichero autorizado en el C-16, entre comillas.

<dispositivo> es un número hexadecimal que indica el dispositivo desde donde se realiza la carga. 1 es cassette, 8 es disco (o 9, A, etc.).

El comando LOAD hace que sea cargado un fichero en memoria. La dirección inicial está contenida en los primeros dos bytes del fichero (un fichero de programa). En otras palabras, el comando LOAD siempre carga un fichero en el mismo lugar de donde se ha salvado. Esto es muy importante trabajando en código máquina, ya que hay muy pocos programas que sean totalmente reubicables. El fichero se carga en memoria hasta que se encuentra el fin de fichero (EOF). Ejemplo:

L"PANTALLA",1

Lee un fichero desde cassette.

L"TANQUE",8

Lee un fichero desde disco.

COMANDO: M (EXAMINAR MEMORIA).

PROPOSITO: Representa la memoria como un vaciado hexadecimal y ASCII, dentro del rango especificado.

SINTAXIS: M[<dirección 1>] [<dirección 2>]

[<dirección 1>] es la primera dirección del vaciado de memoria. Opcional. Si se omite, se representa una página. El primer byte es la última dirección especificada.

[<dirección 2>] es la última dirección del vaciado de memoria. Opcional. Si se omite, se representa una página. El primer byte es el dato de [<dirección 1>].

La memoria se representa en el formato siguiente:

<0310 8B 8C 42 CE OE CE 4C F4:..BN.NLI

El contenido de la memoria puede modificarse mediante el editor de pantalla. Mueva el cursor hasta el dato a modificar, haga la corrección deseada y pulse RETURN. Si se indica una posición RAM errónea o si se hace un intento de modificar la ROM, se representa un indicador de error (?).

Un vaciado en ASCII de los datos se representa en INVERSO (contrariamente al vaciado de otros datos representados en la pantalla) a la derecha del dato hexadecimal. Cuando se encuentra un carácter no imprimible, se representa como un punto inverso (.).

Igual que con el comando Disassemble, puede Vd. hacer paginaciones. Esto se consigue escribiendo M y RETURN. Ejemplo: M 1C00

```
<1C00 41 00 AA AA 00 98 56 45 :A.**..VE
<1C08 41 00 AA AA 00 98 56 45 :A.**..VE
<1C10 41 00 AA AA 00 98 56 45 :A.**..VE
<1C18 41 00 AA AA 00 98 56 45 :A.**..VE
<1C20 41 00 AA AA 00 98 56 45 :A.**..VE
<1C28 41 00 AA AA 00 98 56 45 :A.**..VE
<1C30 41 00 AA AA 00 98 56 45 :A.**..VE
<1C38 41 00 AA AA 00 98 56 45 :A.**..VE
<1C40 41 00 AA AA 00 98 56 45 :A.**..VE
<1C48 41 00 AA AA 00 98 56 45 :A.**..VE
<1C50 41 00 AA AA 00 98 56 45 :A.**..VE
<1C58 41 00 AA AA 00 98 56 45 :A.**..VE
```

COMANDO: > (SIGNO MAYOR QUE)

PROPOSITO: Puede utilizarse para definir de 1 a 8 posiciones de memoria simultáneamente.

SINTAXIS: >dirección byte de datos 1 byte de datos 2...8>

dirección: es la primera dirección de memoria a definir.

byte de datos 1: Datos a introducir en la dirección.

<byte de datos 2-byte de datos 8> son los datos a colocar en las sucesivas posiciones de memoria a continuación de la primera. Opcional. Ejemplo:

>200 08

Coloca un 08 en la posición 2000.

> 3000 23 45 65

Coloca un 23 en la posición 3000, un 45 en la 3001 y un 65 en la 3002.

COMANDO: R (EXAMINAR REGISTROS).

PROPOSITO: Muestra los registros más importantes del 6502: el contador de programa, el registro de estado, el acumulador, los registros índices X e Y y el puntero del STACK.

SINTAXIS: R.

Ejemplo:

```
R
PC   SR AC XR YR SP
1002 01 02 03 04 F6
```

NOTA: El ; (punto y coma) puede utilizarse para modificar el contenido de los registros del mismo modo que se utiliza > para modificar los registros de memoria.

COMANDO: S (SALVAR)

PROPOSITO: Salva el contenido de la memoria en una cinta o disco.

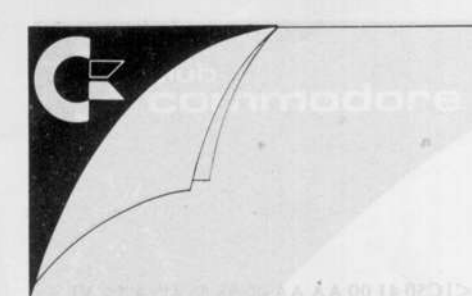
SINTAXIS: S "nombre fichero", <dispositivo>, <dirección 1>, <dirección 2>.

<"nombre fichero"> es cualquier nombre de fichero autorizado en el C-16. Para salvar datos, se debe incluir el nombre de fichero entre comillas.

<dispositivo> El cassette y el disco son los dos dispositivos posibles. Para salvar en cassette, utilice el dispositivo 1. El número de dispositivo de la unidad de disco habitual en el C-16 es el 8. De todos modos, éste puede variar (por ejemplo, al utilizar más de una unidad de disco). Vea su MANUAL DE LA UNIDAD DE DISCOS del C-16.

<dirección 1> es la dirección inicial de la memoria a salvar.

<dirección 2> es la dirección final de la memoria a salvar +1. Se salvan todos los datos hasta aquí, pero sin incluir el byte de datos de esta dirección.



El fichero creado por este comando es un fichero de programa. Los primeros dos bytes contienen la <dirección 1> de los datos. Este fichero puede ser vuelto a llamar mediante el comando L. Ejemplo:

S"JUEGO",8,0400,0C00

Salva la memoria desde $0400 hasta $0BFF pasándola a disco.

COMANDO: T (TRANSFERIR).

PROPOSITO: Transfiere segmentos de memoria desde una a otra área de memoria.

SINTAXIS: T<dirección 1> <dirección 2> <dirección 3>.

<dirección 1> es la dirección inicial de los datos a mover.

<dirección 2> es la dirección final de los datos a mover.

<dirección 3> es la dirección inicial de la nueva posición (donde irán los datos).

Los datos pueden moverse desde una zona baja de la memoria a una alta o viceversa. Se pueden mover segmentos adicionales de memoria de cualquier longitud cualquier número de bytes hacia adelante o hacia atrás.

Nota: El valor de <dirección 3> no debe estar entre <dirección 1> y <dirección 2>.

Ejemplo:

T 1401 1600 1400

Desplaza los datos a partir de $1401, incluyendo $1600, un byte hacia abajo de la memoria.

COMANDO: V (VERIFICAR).

PROPOSITO: Verificar un fichero en cassette o disco con el contenido de la memoria.

SINTAXIS: V <"nombre fichero">, <dispositivo>.

<nombre fichero> es cualquier nombre de fichero autorizado en el C-16.

<dispositivo> es un número hexadecimal indicando en qué dispositivo se encuentra el fichero. El cassette es 1 ó 01, el disco es 8 ó 08, 09, etc.

El comando Verify compara un fichero con el contenido de la memoria. El C-16 responde VERIFYING. Si se encuentra un error, se añade la palabra ERROR. Si el fichero es verificado correctamente, reaparece el cursor parpadeante.

EJEMPLO: V"TRABAJO",8

COMANDO: X (SALIDA AL BASIC)

PROPOSITO: Vuelve al BASIC

SINTAXIS: X.

Cuando se da el comando X, el puntero del STACK del microprocesador se ajusta al valor actual (véase el comando R). Si se modifica éste de cualquier modo, utilice el comando BASIC CLR para restaurar los punteros tras volver al BASIC. ■

Por Pere MASATS

El intérprete de BASIC del COMMODORE 16

El nuevo ordenador doméstico COMMODORE 16 incorpora nuevas instrucciones y comandos de BASIC, en este artículo vamos a analizarlos pues constituyen la característica más interesante de este equipo.

El intérprete de BASIC de COMMODORE 16 incorpora 52 comandos e instrucciones que no existen en otras versiones de BASIC implementadas en otros equipos de COMMODORE, por ejemplo en el COMMODORE 64. Estas nuevas instrucciones son:

AUTO	BACKUP
COLLECT	COPY
DELETE	DIRECTORY
DLOAD	DSAVE
HEADER	HELP
KEY	RENAME
RENUMBER	SCRATCH
BOX	CHAR
CIRCLE	COLOR
DO	LOOP
WHILE	UNTIL
EXIT	DRAW
GETKEY	GRAPHIC
GRAPHIC CLR	IF THEN ELSE
LOCATE	MONITOR
PAINT	PRINT USING
PUDEF	RESTORE
RESUME	SCALE
SCNCLR	SOUND
SSHAPE/GSHAPE	TRAP
TRON	TROFF
DEC	INSTR
JOY	RCLR
RDOT	RGR
RLUM	SGN
ERR$(N)	HEX$
VOL	

Además RESTORE y MID$ funcionan de manera algo diferente.

Esta ampliación del lenguaje se puede analizar reuniendo los nuevos comandos e instrucciones en cuatro grandes grupos:

AYUDAS A LA PROGRAMACION.
NUEVAS INSTRUCCIONES DE BASIC PROPIAMENTE DICHO.
INSTRUCCIONES DE ALTA RESOLUCION Y SONIDO.
MANEJO DE DISCO.

Veamos ahora estas instrucciones:

Ayudas a la programación

Facilitan la entrada de programas en BASIC o en código/máquina. Es habitual que un programa nunca funcione perfectamente a la primera por dos tipos diferentes de errores: los de tecleado de las instrucciones (el típico SINTAX ERROR) y los de concepción del propio programa (el tecleado es correcto, pero una fórmula está mal planteada para realizar un cálculo, por ejemplo), en ambos casos en este grupo de instrucciones se incluyen facilidades a la hora de depurar un programa (librarlo de sus errores).

Las instrucciones de este grupo son:

AUTO.—Da el número de la línea siguiente cuando entramos un programa, de manera que no es necesario teclearlo cada vez.

DELETE.—Permite borrar una sección del programa entre dos números de línea. Aunque el uso de este comando de manera excesivamente alegre suele producir frecuentes disgustos, su utilidad es innegable.

HELP.—El COMMODORE 16 posee un elevado espíritu de colaboración y después de darnos un SINTAX ERROR si tecleamos HELP nos lista en la pantalla la línea donde existe el error y nos indica la instrucción errónea de manera intermitente.

KEY.—Mediante este comando se pueden asignar a las ocho teclas de función secuencias de caracteres de uso frecuente. Por ejemplo, cuando se pone en marcha el equipo si pulsamos la tecla F8, en la pantalla aparece la palabra HELP y la máquina sola pulsa RETURN.

RENUMBER.—Es frecuente que durante la etapa de depuración de un programa sea necesario añadir entre dos líneas más de las que caben entre ellas. Con RENUMBER se pueden cambiar los números de línea de programa para "hacer sitio". De la misma manera cuando el programa funciona perfectamente se pueden numerar las líneas de una en una.

TRON/TROFF.—Se puede ejecutar un programa —mediante estos comandos— de manera que se nos indique en la pantalla qué línea se está ejecutando en cada momento para poder detectar el sitio donde se producen los problemas.

MONITOR.—Mediante este comando el COMMODORE 16 empieza a traba-

jar con lo que se conoce como MONITOR DE CODIGO/MAQUINA. Este es un programa incluido en la ROM del equipo que facilita la entrada de pequeñas rutinas en código/máquina y su correspondiente depuración. Una descripción detallada de las posibilidades del MONITOR se dará en un artículo aparte.

Nuevas instrucciones de BASIC propiamente dicho

En este grupo se incluyen aquellas instrucciones que enriquecen el lenguaje BASIC propiamente dicho, es decir, aquellas que intervienen directamente en la ejecución de los programas.

DO-LOOP-WHILE-UNTIL-EXIT.—Estas instrucciones forman un conjunto muy interesante que facilita la confección de bucles de ejecución condicional. Las dos primeras (DO y LOOP) señalan el principio y final del bucle, respectivamente. Y las tres últimas manejan las condiciones de salida de este bucle. Sus traducciones al castellano pueden esclarecer algo su sentido:

DO=HACER (marca el inicio del bucle).

LOOP=BUCLE (marca el final del bucle).

WHILE=MIENTRAS (MIENTRAS se cumpla la condición, se ejecutará el bucle).

UNTIL=HASTA (HASTA que se cumpla la condición, se ejecutará el bucle).

EXIT=SALIDA (si se llega a ejecutar EXIT, la ejecución del bucle se interrumpe).

Tanto si se interrumpe la ejecución del bucle como si se sale de él, lo que ocurre es que la ejecución del programa pasa a la instrucción que sigue a LOOP.

Para aclarar ideas veamos unos ejemplos:

```
10 DO UNTIL A    29
20 A=A+1
30 PRINT A
40 LOOP
50 PRINT A+50
```

En este caso la secuencia de líneas 10—40 se ejecutará hasta el momento que A tome un valor superior a 29, entonces se saldrá del bucle y se pasará a ejecutar la línea 50.

Si tenemos:

```
10 DO WHILE A    30
20 A=A+1
30 PRINT A
40 LOOP
50 PRINT A+50
```

Se ejecutará el bucle MIENTRAS A sea inferior a 30, en cuanto deje de cumplirse esta condición se transferirá la ejecución a la línea 50.

Para EXIT podemos hacer:

```
10 DO
20 A=A+1
30 IF A=30 THEN EXIT
40 LOOP
50 PRINT A+50
```

En cuanto se cumpla la condición A=30 en la línea 30, se ejecutará la línea 50.

GETKEY.—Esta instrucción permite entrar caracteres desde el teclado de uno en uno:

```
10 GETKEY A$
```

Es exactamente igual a:

```
10 GET A$ : IF A$="" THEN 10
```

Tanto en una versión como en otra el programa esperará a que se pulse una tecla, en cuanto se realice esta acción A$ contendrá, en forma de cadena de un solo carácter, la tecla pulsada.

IF...THEN... ELSE...—En el COMMODORE 16 se ha ampliado la instrucción estándar de BASIC IF... THEN y se le ha incorporado la opción ELSE. Normalmente, SI (IF) se cumple la condición descrita entre IF y THEN ENTONCES (THEN) se ejecuta la instrucción que sigue a THEN (o serie de instrucciones hasta el fin de la línea). En caso de no cumplirse la condición el programa sigue ejecutándose a partir de la línea siguiente. La versión del COMMODORE 16 es igual por lo que se refiere a IF... THEN... si la condición se cumple. Lo que ocurre es que si no se cumple se ejecutan las instrucciones que siguen a ELSE.

PRINT USING y PUDEF.—PRINT USING es una ampliación de PRINT, su utilización es primordialmente para controlar de forma eficaz el formato de los datos de salida de un programa, tanto en pantalla como en la impresora. En aplicaciones de gestión es muy molesto que las columnas de números que entrega un programa como resultados no estén alineadas en sentido vertical (las unidades exactamente debajo de las unidades, las decenas de las decenas, etc....), también mediante PRINT USING se pueden truncar los números a un número determinado de decimales, eliminar, sustituir o incorporar ceros no significativos, incorporar el signo, etc. PUDEF complementa a PRINT USING permitiendo —por ejemplo— sustituir el punto decimal anglosajón por la coma utilizada en nuestro país. PRINT USING y PUDEF ahorran al programador mucho trabajo de escritura de subrutinas y simplifican los programas.

RESTORE.—Esta instrucción funciona esencialmente igual a otras versiones de BASIC, la única diferencia es que se puede hacer que el puntero de los DATAS se restablezca en un número de línea determinado, así es posible leer (READ) una serie de DATAS situados al final del conjunto sin tener que leer los primeros. O dicho de otra manera: podemos escribir los DATAS en nuestro programa en el orden que nos sea más conveniente, mediante RESTORE XXXXX podemos leerlos como queramos.

TRAP y RESUME.—Mediante estas dos instrucciones de BASIC es posible que el programador tenga un control eficaz de su programa si se produce un error. Así es posible que el programa siga funcionando en una rutina especialmente escrita y designada en la instrucción TRAP.

Para hacer posible el tratamiento de los errores existen tres variables que el ordenador utiliza para transferir al programa la información necesaria sobre el lugar del error y su naturaleza, estas variables son: EL, da el número de la línea donde se ha detectado el error; ER, es el número del error y ERR$, es una matriz que contiene los mensajes de error. Así PRINT ERR$(ER) dará el error producido y en función de su contenido el programa realizará una acción. Podríamos decir que mediante TRAP XXXXX se llama a una subrutina que empieza en la línea XXXXX y que se ejecutará cada vez que se produzca un error, en vez de utilizar RETURN al final de esta subrutina se emplea RESUME. RESUME sin más intentará ejecutar de nuevo la línea donde se ha producido el error, RESUME NEXT ejecuta la línea siguiente a la que produjo el error y RESUME YYYYY pasará el control del programa a la línea YYYYY.

Funciones numéricas y de manejo de cadenas.

JOY.—Esta función nos entrega la posición de las palancas de juegos (JOYSTICKS).

SGN.—Si queremos saber el signo de la cantidad almacenada en una variable, SGN(X) nos entrega el valor de este signo sin necesidad de manejar el número.

DEC y HEX$.—Si tenemos un número hexadecimal en la cadena alfanumérica X$ y hacemos X=DEC(X$), en X tendremos el equivalente decimal de este número. HEX$ efectúa la operación contraria: convierte un número en base 10 a otro en hexadecimal.

MID$.—Esta función de manejo de variables de cadena en el COMMODORE 16 tiene una utilización que no existe en otras versiones de BASIC. La diferencia consiste en que MID$ puede estar en la parte izquierda del signo "=", es decir, se puede "meter" en una cadena, otra de menor longitud en las posiciones especificadas por los parámetros de esta función.

INSTR.—Se utiliza para saber si una cadena forma parte de otra, en caso afirmativo esta función entrega el número de carácter donde empieza la primera dentro de la segunda.

El mes próximo seguiremos este análisis del BASIC del COMMODORE 16 viendo el capítulo de instrucciones gráficas y los comandos de disco.



Las Aventuras de KBIT KBITA
REMY Y REMO
por ERRAZQUIN
¡Hagan Humo Señores..!
OH NOOO... DOS HISTORIETAS SEGUIDAS OLIENDO A CHAMUSQUINA ES DEMASIADO
COMMODORE WORLD



NO SI... NO HAY FUEGO. SOY YO... ¡EJEM!
¿?



AAAAHHH



ARRFFF
GLLLL
GLLLL
VAMOS VAMOS TAMPOCO ES PARA PONERSE ASI...
SI SOLO SOY UN GRAFICO
POM POM



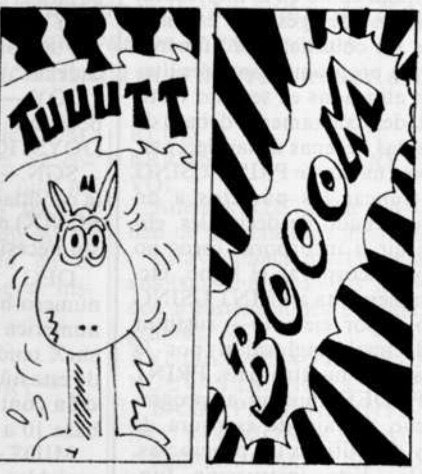
HECHO PIXEL A PIXEL HECHO DE QUEEE...
DE PIXELS... UN PUNTO LUMINOSO... CADA CARACTER, POR EJEMPLO ESTÁ FORMADO POR UN CONJUNTO DE PUNTOS LUMINOSOS O PIXELS. SI LOS COMBINAS BIEN... PUEDES HACER UN GRAFICO O UN DIBUJO
PIXEL
CARACTER
O PIXEL ENCENDIDO
● APAGADO



LA PANTALLA SE ILUMINA O NO PUNTO A PUNTO, (PIXEL) Y ASI ME HICIERON A MI ... ¿CURIOSO NO?
SIIIII PERO TU ERES UN... DRAGON ENORME



BUENO... SIII... BIEN MIRADO QUIZAS ME DIBUJARON ALGO GORDO ¡¡¡



TUUUTT
BOOOM



¡HORROR! ... SEGURO QUE SON ELLOS ESTOY PERDIDO... ... TENGO MIEDOOOO
LEVANTATEEE LEVANTATEEEE TE DIGOOO
POM POM



DEBE DE SER KBITA HACIENDO PRACTICAS DE TROMPETA... Y... ¿PERO TE QUIERES LEVANTAR DE UNA VEZ?
NO PODRE RESISTIRLO LO SE



ESTOY TAN TERRIBLEMENTE ASUSTADO...
CRUNCH
AGGG

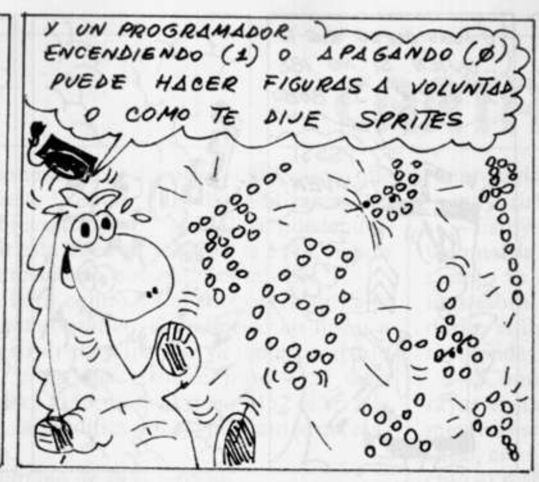


PARA UN SPRITE LA VIDA NO ES FACIL
UN QUEEE



NO SABE LO QUE ES... JUAAA... JUAAA... JUAAAA
PARAAAA
BOOM BOOM BOOM


TE LO VOY A CONTAR... YA SABES QUE UNA LETRA O UN NUMERO SON EL REALIDAD UNOS CUANTOS PIXELS O PUNTOS DE LUZ EN LA PANTALLA...
CON LUZ
SIN LUZ
MIRA EL NUMERO 7 POR EJEMPLO


Y UN PROGRAMADOR ENCENDIENDO (1) O APAGANDO (Ø) PUEDE HACER FIGURAS A VOLUNTAD O COMO TE DIJE SPRITES


COMO LOS PUEDES MOVER A VOLUNTAD POR LA PANTALLA... PUEDES HACER BONITOS JUEGOS ANIMADOS
FIUUU
BZUMM
BOOOM


JOPE TU... QUE INTERESANTE ... ¿ PERO Y ENTONCES POR QUÉ HUYES?




AUUUUUUU


ME QUIEREN BORRAR DE LA MEMORIA... SNIF... SNIF. TIENES QUE AYUDARME
GLLU


ADEMAS.. COMO HABRAS PODIDO OBSERVAR SOY UN DRAGON MUY CARIÑOSO..
FSSS


... UN MOMENTO. ¿SOY UN DRAGON O UN GUSANO? ... SI SE ATREVEN A VENIR LOS PULVERIZO Y...
POF


TOC
TOC
CRIIIICC


... SOY UN GUSANOOOOO... TENGO MIEDO
BOOM
BOOM
AHHHHH


ABRAAANNN
TIENES QUE ESCONDERME ¿ ME OYES? ¿ EH?


VALE.. VALE.. DE ACUERDO PERO NI SALTES.. NI TE PONGAS BESUCON.. ¿ VA?


ESTA BIEN.. ¿ DONDE DEMONIOS ESTÁ LA LAGARTIJA ESA?
BLAM


PUES AQUI NO ESTÁ... POR CIERTO VAYA CUADRO MAS FEO...
ES... ES MI ABUELO
PUES MIRA QUE ES FEO EL TIO
EJEM.. SII...
ERRAZQUIN/85


... BIEN.. ES IGUAL... TU ME ACOMPAÑARAS A VER AL EMPERADOR
¿YO?


¿PUES CLARO QUE TU ..QUIEN SI NO IBA A SER? .. SO BOBO
PSSST ¡JOVEN!


FLOSS


MAMAAAAA
JI JI


¿Y NO SERIA MEJOR QUE LE PREGUNTASEMOS QUE ES LO QUE QUIERE?


PARA... PARAAA
BOOM BOOM


..Y AHORA ESCUCHAME.. VOY A IR A VER QUE QUIERE... Y TU TE VAS A ESTAR AQUI... PERO QUIETECITO...
GRRRRR


SI NO ARREGLO PRONTO ESTO, ES CAPAZ DE SEGUIR DANDO SALTOS Y ROMPER LA HISTORIETA.... ...¡EH... SARGENTO... PAREEE..!


DEJEMEEE... QUE ME DEJE LE DIGO.. VOY A PEDIR LA BAJA EN EL EJERCITO... ..NO... SUELTEEE
SOLO QUIERO QUE ME DIGAS PARA QUE QUIERES A MI AMIGO


PUES PARA... ZIU.. ZIU... BLA, BLA, BLA ¿Y AHORA ME PUEDO IR?


..NO TE QUIEREN BORRAR SINO QUE VAYAS A LA RAM PARA UN JUEGO CON UNA DRAGONA...
POM POM

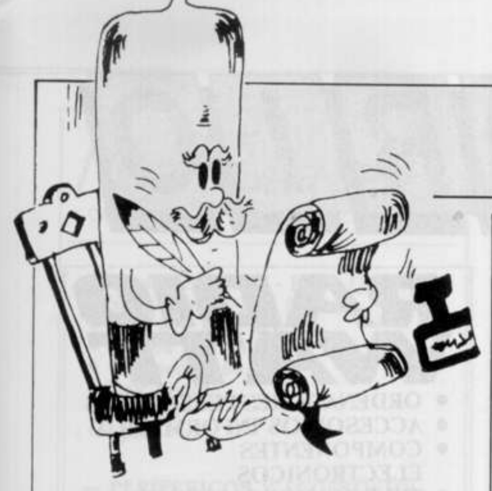

YUPPI!!!!! ... UNA DRAGONA... VOY VOLANDO A VERLAA
BOOM BOOM


PUFFFF... BUENO YA ACABO TODO POR FIN...

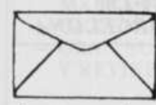

PASOOOO... PASO LIBREEE
BORROMBLOM
BOOOMMM


A MI NO ME DIGA NADA YO YA NO SOY DEL EJERCITO... SOCORROOOO
BORREMEEE... SE LO RUEGO... ¿ES QUE NO LA HA VISTO?
YUUUHH... "AMORSITO" ..VEEEN...
FUFF
ERRAZQUIN/85

CARTA BLANCA



PROCESADOR DE TEXTOS



Con agrado he visto publicada mi colaboración en el nº 13 (marzo-85) "Modificaciones en el procesador de textos para C-64", pero por desgracia, debido a un error, envié una versión que no era la definitiva, de manera que el funcionamiento no es correcto. El problema se solventa modificando la línea 1149 de forma que quede así:

1149 IF Q$="C" THEN N1$=LEFT$ (N1$,16):GOTO 1162

Esta línea limita el nombre del fichero a 16 caracteres y salta a la apertura y lectura del fichero en cinta, proceso que sin esta corrección se realizaba de manera incondicional, cualquiera que fuera la tecla pulsada. Utilizando las abreviaturas del BASIC, se puede añadir esta línea (sin el IF) a la 1148, siendo el resultado obtenido, el mismo.

Para colmo, también cometí un fallo mecanográfico, al enumerar las líneas a destruir para los que ya tenían grabado el programa. Donde puse 1152 **debe poner 1155**, pues la línea 1152 es creada en la modificación y, por tanto, necesaria.

Mariano de Blas Gracia
C/ Trabau, 61 - 1º-2.ª
08031 Barcelona

LOAD & RUN



Ante todo mis felicitaciones porque creo que hacen una revista muy lograda y que nos es muy útil.

Les escribo esta carta porque tengo una serie de dudas que les quisiera consultar:

—¿Para qué sirve la secuencia SHIFT-RETURN?

—¿Cuántos sprites es posible utilizar a la vez? En el manual dice que son ocho, pero he visto juegos en los que los "marcianitos" son más de ocho, si bien la mayoría son iguales.

—Estoy haciendo un programa de Geometría que pienso mandarles en cuanto lo acabe, pero como no tengo impresora se me hace imposible mandarles el listado que piden. ¿Es necesario mandarlo o basta sólo con el cassette?

—¿Cómo se consiguen los dibujos o rayas en la presentación de programas mientras se cargan? De ser muy complicado, creo que podrían dedicar un artículo al respecto, opino que sería interesante.

Fermín Bernaus Berraondo
C/ Juan de Garay, 5-3º A
20008 San Sebastián

—La secuencia SHIFT-RETURN tiene el mismo efecto que RETURN, sólo que no se ejecutan las órdenes.

—Son ocho. Que en un juego aparezca un "marcianito" no quiere decir que sea un sprite, puede ser un gráfico hecho con caracteres personalizados que se parezca mucho a un sprite.

—Si no tienes impresora no es necesario que mandes un listado, lo que sí que es imprescindible es la cinta cassette o el disco.

DIRECTORIO



- Ordenadores personales Hard y Soft.
- Cursos de Basic.

RENOVACION EN MARCHA, S.A.
OFICINAS
C/. Espronceda, 34-2º int.
28003 MADRID
Teléfono **(91) 441 24 78**

REM SHOP 1
C/. Galileo, 4 - 28015 MADRID
Teléfono **(91) 445 28 08**

REM SHOP 2
C/. Dr. Castelo, 14 - 28009 MADRID
Teléfono **(91) 274 98 43**

REMSHOP-3
C/. Modesto Lafuente, 33
28003 MADRID
Teléfono **(91) 233 83 19**

REM SHOP - BARCELONA
C/. Pelayo, 12 - Entresuelo J
Teléfono **(93) 301 47 00**

REM SHOP - LAS PALMAS
Gral. Mas de Gaminde, 45
Teléfono **(928) 23 02 90**
(Inauguración) 25/2/85

---

casa de software s.a.

c/ aragón, 272, 8º, 6.ª
tel. 215 69 52 - 08007 barcelona

- **Software profesional para C-64**
- **Con distribución productos DIGITAL RESEARCH**

SOFTWARE ESPAÑA

Avda. de Arteijo, 19
14004 La Coruña
Teléf. 25 51 72

Especializados en software para Commodore-64 Spectrum y MSX

COMPARE LOS PRECIOS

COMMODORE 64	54.500
DATASSETTE	10.000
FAST-TURBO MENUE	8.500
SOLO FLIGHT	3.800
COLOSSUS CHESS 2.0	2.800
ZAXXON	2.300
TOTAL	81.900

POR SOLO 59.900 Ptas.

Modem completo desde	16.000
Commodore SX-64	180.000
Floppy 1541	59.900
Commodore 64	49.900
Impresora MP-2080	63.500
Amstrad Fósforo verde	76.000
Amstrad Monitor color	115.000

ASTOC DATA
Apartado de Correos, 695
SANTIAGO DE COMPOSTELA
Tel. (981) 59 95 33



HACEMOS FACIL LA INFORMATICA

●SINCLAIR ● SPECTRAVIDEO
●COMMODORE ● DRAGON
●AMSTRAD ● APPLE
●SPERRY UNIVAC

Modesto Lafuente, 63 Telf. 253 94 54 28003 MADRID	Colombia, 39-41 Telf. 458 61 71 28016 MADRID
José Ortega y Gasset, 21 Telf. 411 28 50 28006 MADRID	Padre Damián, 18 Telf. 259 86 13 28036 MADRID
Fuencarral, 100 Telf. 221 23 62 28004 MADRID	Avda. Gaudí, 15 Telf. 256 19 14 08015 BARCELONA
Ezequiel González, 28 Telf. 43 68 65 40002 SEGOVIA	Stuart, 7 Telf. 891 70 36 ARANJUEZ (Madrid)

COMPAÑIA ESPAÑOLA DE SUMINISTROS ELECTRONICOS (C.O.S.E.S.A.)

ORDENADORES PERSONALES

—SINCLAIR —SPECTRA VIDEO
—COMMODORE
—LASER —SONY

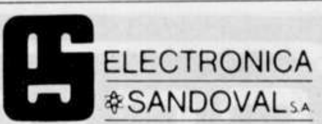
C/ Barquillo, 25
Telfs.: 222 69 49-232 36 44-231 29 18-221 55 07
28004 MADRID

RADIO WATT

- ORDENADORES PERSONALES
- ACCESORIOS INFORMATICA
- COMPONENTES ELECTRONICOS
- TELECOMUNICACIONES

Paseo de Gracia 126-130
Tel. 237 11 82. 08008 BARCELONA*

CENTRO DE INFORMATICA
Las Rozas - Majadahonda
EMPEZAMOS
Cursillos en BASIC cada 15 días
Tenemos todo lo referente al COMMODORE 64
Teléfono: 637 31 51



DISTRIBUIDORES DE:

ORIC-1
CASIO FP200
ROCKWELL-AIM-65
VIDEO GENIE-EG-2000
CASIO FX-9000P
SINCLAIR SPECTRUM
OSBORNE 1
DRAGON-32
NEW BRAIN
EPSON HX-20

C/SANDOVAL, 3, 4, 6 MADRID 28010
TELEFONOS 445 18 33/18 70
TELEX 47784 SAVL-E

VENTAMATIC

- VEN A CONOCERNOS. Somos los SUPER-ESPECIALISTAS del COMMODORE 64 y lo tenemos TODO para tu COMMODORE 64 (incluyendo lo último en accesorios y programas musicales y MIDI). SOLICITA CATALOGO COMPLETO.
- VENTAMATIC · c· Córcega, 89, entlo. 08029 BARCELONA · Tel.: (93) 230 97 90. Metro ENTENZA Línea V. Bus: 41, 27, 15, 54, 66.
- Departamento de Envíos y Venta por Correo: VENTAMATIC · Avda. de Rhode, 253 · ROSES (Girona) · Tel.: (972) 25 79 20.

—Utilizando las interrupciones, como se explicó en el número diez de COMMODORE WORLD. Generalmente hay que hacer una rutina de carga completa en lenguaje máquina.

COMMODORE 16



Pienso adquirir un C-16, pero tengo unas dudas que me quedaron después de leer la reseña del número 13 de esta revista.

1.—¿Los programas del VIC-20 o del C-64; al nuevo 16 son adaptables?

2.—Tienen en proyecto publicar el mapa de memoria de este ordenador, para hacer adaptaciones de estos programas que incluyen sentencias POKE y PEEK?

Como pueden ver, lo que más me preocupa es la falta de software para el C-16, el cual, opino, es el ordenador con más futuro para, los que como yo, desean aprender a programar estas máquinas y no disponen de muchos ingresos.

Jesús E. Jiménez
C/ General Aranda, 8
020 Minaya (Albacete).

1.—Por supuesto, en un número de estos publicaremos los cambios al igual que hicimos con el VIC y el 64.

2.—Todavía no hemos visto el mapa, pero intentaremos conseguirlo.

DESIERTO VIC-20



El programa Vídeo Casino. Aventuras en el desierto del nº 4, que se corrigió en el nº 8, no logro ponerlo en **marcha**, por lo que le rogaría me indicase si hay algún error en el mismo.

Enrique Hausmann
C/ Constitución, 3
07001 Palma de Mallorca.

La corrección del número ocho hacía que pudieran verse las ruedas del jeep, por lo demás el programa está bien. No entendemos qué quieres decir al no poder "ponerlo en marcha". Si lo que no te funciona es el RUN, comprueba la memoria que tienes conectada al ordenador y que la carga desde cinta se haya realizado correctamente.

MULTICOLOR O ALTA RESOLUCION



Queridos amigos, os mando dos preguntas, aunque lo segundo sea más bien una sugerencia. Ahí van: (son para el VIC 20).

1º ¿Cómo se pueden colorear bit a bit los caracteres definidos por el usuario? Por favor, explicádmelo muy clarito, porque si no, no me entero.

2º ¿Por qué en los programas que aparecen en la revista, y que se mueven con el teclado, no incluís vosotros aparte una pequeña rutina para que se pudiesen mover con joystick, o viceversa? (Naturalmente, esto sólo se podría hacer en los programas de juegos).

Alvaro Minguito
C/ Numancia, 30-7º C
Madrid

1.—Bit a bit no se puede, sí por parejas y utilizando sólo tres colores. El artículo "Vic Multicolor" de este número puede que te aclare cómo se hace.

2.—Ya lo hemos hecho con algunos, e intentaremos seguirlo haciendo.

CARACTERES PERSONALIZADOS Y SUPER-EXPANDER



Os escribo para haceros una pequeña pregunta, veréis, resulta que yo tengo un editor de caracteres para el VIC-20 copiado de la Commodore World nº 7 (pág. 46), pero los caracteres funcionan con el VIC sin ampliación, me gustaría saber si hay alguna manera de que se puedan imprimir los caracteres en la pantalla del VIC con el super-expander conectado en el modo gráfico, gracias.

Alvaro Nieto Pérez. Nº suscriptor 2882
C/ Blas de Otero, 61-10º C.
Telf.: 447 68 76
48014 Bilbao

Hemos recibido muchas cartas preguntando lo mismo que tú. Que sepamos, no hay otro método para hacerlo que no sea dibujar el carácter pixel a pixel, lo cual además de ser lento es bastante complicado.

ERRORES POR TODAS PARTES



Soy suscriptor de vuestra revista desde el nº 12. Me apunté a ella ante la necesidad de tener un asesoramiento de mi lento y solitario trabajo en el aprendizaje.

Poseo un VIC-20 y aunque hace ya un año que lo tengo, mis conocimientos de programación son todavía muy escasos. Si bien mis estudios no me dejan demasiado tiempo para dedicarme a mi ordenador (estudio Ingeniería Técnica Industrial), procuro entender y ejecutar los programas que aparecen en la revista. No obstante parece ser que no tengo mucho éxito y, aunque ya voy aprendiendo, no consigo llevar a cabo ninguno de los dos programas de los que aparecieron en el mencionado nº 12, de los que he intentado. Mis dificultades están en:

1º) El programa "Cuevas de Alfa

SEAMOS PREGUNTONES ¿?



Ceti". Después de listarlo cambia la pantalla a blanco y aparece: "Un momento...". Continúa con: "Error en los datas" y se para el programa. Por mucho que repaso no encuentro ningún error en los datas que listé.

2º) El programa "Llançador". La línea 110 no aparece en el listado, y cuando intento introducirla sólo consigo hacerlo hasta el último POKE 36. Pero a partir de ese 6 ya no puedo grabar el resto de la línea pues no entra, incluso copio nuevamente la línea entera y pulso RETURN, sin que esta nueva línea sustituya a la anteriormente escrita.

Me gustaría también que me indicárais brevemente las principales características técnicas del VIC: capacidad de memoria, etc., pues en mi entorno al menos, parece que hay un poco de confusión.

Fco. Ginés Pastor Través
C/ Eduardo Dato, 45-9º A. Telf.: 58 01 76
41005 Sevilla

1.—Mira a ver si el número que comprueba los datas (línea 60) es correcto. Si no lo es probablemente que te has dejado alguna coma o la has cambiado por un punto o porque has tecleado una "o" en vez de un cero.

2.—El VIC sólo acepta líneas de 88 caracteres de longitud (el C-64 sólo 80). Esta línea es más larga de lo permitido por lo que tienes que utilizar las abreviaturas de los comandos POKE y NEXT. POKE abreviado es P[SHIFT O] y NEXT es N[SHIFT E].

El VIC "a secas" tiene 3.5 Kbytes de memoria libre y es ampliable a 32K, de los que sólo unos 29.000 se pueden usar en Basic. También hay unas tarjetas de 64K que actúan a modo de ficheros.



"Commodore World"
aparte de venderse generalmente en kioscos, se encuentra asimismo a la venta en las siguientes distribuciones de Commodore y librerías.

ALICANTE
- Basic. c/San Mateo, 11. Teléfono. (965) 21 10 41.
- Casa Wagner. c/Juan Carlos I, 37. Elda. Teléfono: (965) 39 03 96.
- Ferretería Progreso. c/General Jordana, 28. Elda. Teléfono: (965) 38 11 45.

ASTURIAS
- Cuadrado Informática. c/Toreno, 5. 33004 Oviedo. Tel.: (985) 24 06 21.

BADAJOZ
- Control Sistemas. Avda. Santa Marina, 25A. Teléfono: (924) 25 88 00.

BARCELONA
- Catinsa Informática. c/de L'Esglesia, 15. Teléfono: (93) 784 27 17. Terrasa.
- Comercial Clapera. c/Mariano Maspons, 4. Granollers. Teléfono (93) 870 45 42.
- Computerhard c/s. Jaime, 48. Granollers. Teléfono: (93) 870 09 19.
- Electrodomésticos Mirambell. c/Rabal, 45. San Sadurni d'Anoia. Teléf.: (93) 891 11 34.
- Librería Emilia Pérez Radua, c/Mayor, 35. Castellar del Vallés. Teléfono: (93) 714 89 51.
- Librería Michel. Rda. Guinardo, 1. Sardañola.
- Mozart. c/Jaime I, 145. Mollet. Teléfono: (93) 593 75 01.
- Radio Watt. Paseo de Gracia, 130. Teléfono: (93) 237 11 82.
- Register Lately. c/Balmes, 297. ppal. 2.ª Teléfono: (93) 200 18 99.
- Sonimóvil. c/Alcalde Armengou, 53. Manresa. Teléfono: (93) 873 78 17.
- Suministradora Z, S.A. Av. Barbera, 49-51. Sabadell. Teléfono: (93) 710 56 66.
- Tronic. Bigay, 11-13. Tel.: (93) 212 85 96.

BILBAO
- Bilbomicro, S.A. c/Aureliano Valle, 7. 48010 Bilbao. Teléfono: (94) 443 43 51.

BURGOS
- E.I.S.A. c/Madrid, 4. Teléfono: (947) 20 46 24.

CACERES
- Electrónica Cáceres. c/Badajoz, 45. Teléfono: (927) 248 899.
- Informática Vivas White. San Pedro, 8. Teléfono: (927) 24 40 96.

CADIZ
- Video Computer. Comandante Gómez Ortega, 59. Algeciras. Tel.: (956) 65 39 02.

CASTELLON
- Horizontes. c/Navarra, 76.

CIUDAD REAL
- Electrónica Turrillo. c/Pedrera Baja, 7. Teléfono: (926) 22 38 67.

LA RIOJA
- Librería Sánchez Ochoa. c/Sagasta, 3. Teléfono: (941) 258 622 Logroño.

LA CORUÑA
- Cetronic, S.L. c/Palomar, 22 Bajo. Teléfono: (981) 27 26 54.
- Gesty Computer. Avda. Romero Donallo, 25. Tel.: (981) 59 87 54. Santiago de Compostela.
- Sanlusa, S.L. San Luis, 46 al 50. Teléfono: (981) 23 07 49.

GERONA
- Digit Informática. c/Avda. 11 septiembre, 7. Olot. Teléfono: (972) 26 94 01.
- Microchip. c/Aigua, 3. Olot. Tel.: 26 36 63
- Regiscompte, S.A. c/Emilio Grahit, 17 Bis. Teléfono: (972) 21 99 88.

GRANADA
- Rafael Moreno Torres. c/Recogidas, 24. Edif. Castro. Teléfono: (958) 26 20 50.

HUELVA
- Computerlog, S.A. c/Tendaleras, 15. Teléfono: (955) 25 81 99.

IBIZA
- Kelson. Plaza España, s/n. San Antonio Abad. Teléfono: (971) 34 13 09.

LERIDA
- Albareda. c/Carmen, 19. Tel.: (973) 31 04 02. Tárrega.

LEON
- Logdata. c/Burgo Nuevo, 4. Teléfono: (987) 20 42 89.
- MicroBierzo. c/Carlos I, 2. Teléfono: (987) 41 74 21 (Ponferrada).

MADRID
- Key Informática. c/Embajadores, 90. Teléfono: (91) 227 09 80.
- Librería García Peña. c/Cavanilles, 52.
- Micromundo, S.A. El Zoco. Majadahonda. Teléfono: (91) 638 13 89.
- Micros Garden. c/Francisco Silvela, 19. Teléfono: (91) 401 07 27.
- Remshop. Galileo, 4. Tel.: (91) 445 28 08.
- Winkel-Microsoft M-2 La Vaguada. Local B 82-83. Teléfono: (91) 730 26 22.

MALAGA
- Informática Martínez, S.A. c/Cristo de la Epidemia, 90. Teléfono: (952) 26 37 68.

ORENSE
- Bermello. c/Gral. Franco, 123.

PAMPLONA
- Itar Computers, S.A. c/Alfonso el Batallador, 16 (Trasera). Tel.: (948) 27 64 04.

PONTEVEDRA
- La Boutique del Ordenador. Velázquez Moreno, 1 Bajo. Vigo-1. Teléfono: (986) 22 45 36.

SALAMANCA
- Info-Gest. c/Ronda del Corpus, 2-1º centro. Teléfono: (923) 21 59 93.

SAN SEBASTIAN
- Dommicro, S.A. c/Arrasate, 6. Teléfono: (943) 42 35 10.

SANTANDER
- Librería Hernández. San Francisco, 15. Teléfono: (942) 22 53 30.

SEVILLA
- Microtienda-1, S.A. c/Aceituno, 8. Teléfono: (954) 37 85 57.

TARRAGONA
- Oficomplet. Plaza de la Cinta, 6. Tortosa. Teléfono: (977) 44 14 50.

VALENCIA
- Ineleksa. c/Remedio, 38-bajo-dcha. Sagunto. Teléfono: (96) 266 48 64.
- Librería Mayte. c/Pintor Benedito, 3. Teléfono: (96) 325 28 83.

ZARAGOZA
- ADA Computer. Centro Independencia. Pº Independencia, 24-26. Tel.: (976) 29 85 62.
- Bazar Runa. c/Duquesa Villahermosa, 3. Teléfono: (976) 35 09 48.

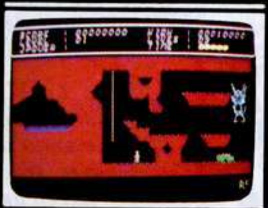
microbyte PRESENTA... SENSACIONALES PROGRAMAS PARA commodore 64

FIGHTER PILOT



Como piloto de una sofisticada aeronave, se te ha encomendado la defensa de las cuatro bases más importantes. Los diferentes niveles de dificultad los podrás superar con tu destreza y la ayuda de la computadora de vuelo, radar, etc. **P.V.P.: 2.200 pts.**

CAVE FIGHTER



Apasionante recorrido a través de 31 cavernas de que consta el juego. Diferentes niveles de dificultad. **P.V.P.: 1.650 pts.**

ANCIPITAL



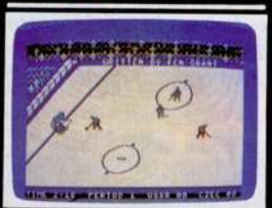
Frenética y sicodélica aventura en la que Ancipital hara volar a sus enemigos cuando estos intenten interponerse en su camino a través del laberinto. **P.V.P.: 1.650 pts.**

ANT ATTACK



El objeto del juego es la supervivencia de una raza advenediza en la ciudad tridimensional y fortificada de las hormigas gigantes. **P.V.P.: 1.750 pts.**

SLAP SHOT



Espectacular partido de Hockey sobre hielo. Posibilidad de elegir diferentes equipos con diversos niveles de dificultad. **P.V.P.: 1.750 pts.**

POLE POSITION



El más atractivo recorrido a través de los diferentes y más famosos circuitos de la F-1. Habilidad y destreza serán necesarias para situar tu bólido en la POLE POSITION. **P.V.P.: 2.400 pts.**

CLIFF HANGER



El Cañón del Colorado será testigo de la detención, por parte de nuestro héroe Clift, de la banda, que ha asolado el Oeste. Diferentes niveles de dificultad. **P.V.P.: 1.650 pts.**

GRYPHON



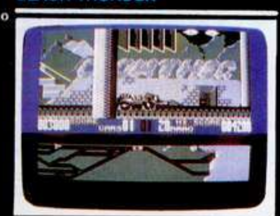
A traves de un mundo subreal con peligrosas tierras oscuras, GRYPHON se tiene que abrir camino para poder llegar a su nido. **P.V.P.: 1.750 pts.**

ZAGA MISSION



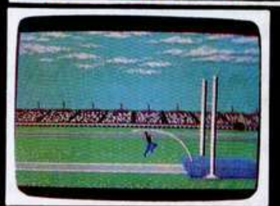
Siente la sensación de manejar un helicóptero en un laberinto tridimensional. Solo con una absoluta concentración podrás completar los diez niveles de juego. **P.V.P.: 1.650 pts.**

BLACK THUNDER



La trepidante velocidad de la nave hace muy difícil detectar con la suficiente antelación los misiles que quieren impedir un final sin contratiempos. **P.V.P.: 1.750 pts.**

SUMMER GAMES (2 Cassettes)



Conjunto de ocho pruebas atléticas en la competición más afamada de los juegos de verano. Posibilidad de elección de los ocho participantes en busca de las medallas que el ordenador otorga a los mejores. **P.V.P.: 3.400 pts.**



Envíenos a **MICROBYTE,**
Pº Castellana, 179-1.º. 28046-Madrid

Nombre
Apellidos
Dirección
Población
D.P. Teléfono

Incluyo talón nominativo ☐
Contra-Reembolso ☐

ENVIOS GRATIS

Juego	Precio	TOTAL
PRECIO TOTAL PESETAS		

Pedidos por Teléfono **91-442 54 33**

Magia



La MAGIA son trucos, la MAGIA es divertida.

La MAGIA es hacer lo que nadie se ha atrevido y resulta ser la fuente más completa de información para la informática práctica.

La MAGIA es una sección mensual llena de consejos, trucos, de esto y aquello del mundo del software, hardware y aplicaciones, trucos descubiertos por los demás que hacen que la informática sea más fácil, más divertida o más animada.

MAGIA habla de ideas sencillas, programas de una sola línea, subrutinas útiles, hechos de informática poco conocidos y otras cosas de interés.

Los trucos de magia enviárnoslos comprobados, pues hay varios incorrectos. ¡Ah! y no nos mandéis trucos repetidos... ¡Listos!

MAS PROGRAMAS DE UNA LINEA

En la página 33 del número 1 de "Commodore World" se publicó un divertido programa de una línea con 76 caracteres. He aquí un programa que hace lo mismo, pero con 51 caracteres solamente.

```
1 PRINT"[RVSON][SPC][CRSRL]"MID$("[CRSRU][CRSRD]
[CRSRL][CRSRR]",RND(1)*4+1,1)""*[CRSRL]";:A=2↑2:RUN
```

En la página 51 del núm. 7 se daba un programa de una línea con 69 caracteres para pasar de decimal o binario. Por cierto que al autor se le olvidó decir que sólo vale hasta N=255. Aquí va un programa que hace lo mismo, pero con 43 caracteres solamente:

```
1 INPUTD:FORI=0TO7:PRINT"[CRSRU][INST]"DAND1:
D=D/2:NEXT
```

Un programa con 56 caracteres, pero más general, ya que es válido hasta D=2↑32=4294967296, y más elegante, puesto que no hay ceros a la izquierda, sería el siguiente:

```
1 INPUTD:FORC=0TO0:C=—INT(D/2):PRINT"[INST]"
D+2*C"[CRSRU]":D=—C:NEXT
```

Y ya, para completar, así se haría el paso de binario a decimal, válido hasta mil millones (decimal):

```
1 INPUTB$:FORI=1TOLEN(B$):D=2*D+VAL(MID$(B$,I,1)):
NEXT:PRINTD
```

Ernesto Ferrón de la Fuente
Aller, 31 B, 2º Dcha.
Mieres (Asturias)

UNA "Ñ" PARA EL VIC

Se trata de aprovechar la posibilidad de construir nuestro propio juego de caracteres en el VIC-20 para disponer de la letra "Ñ".

La rutina de abajo es para el VIC-20 sin expansión de memoria y la Ñ queda situada en el lugar de ←. Si queréis situarla en otro sitio no tenéis más que cambiar el 31 de la línea 2 por el código del carácter que sea.

```
1 FORJ=0TO511:POKE7168+J,PEEK(32768+J):NEXT
2 FORK=7168+31*8TO7168+31*8+7:READY:POKEK,V:NEXT
3 POKE52,28:POKE56,28:POKE36869,255
4 DATA90,66,98,82,74,70,66,0
```

César Alvarez

HES-MON

El otro día, leyendo vuestra revista en la sección de "Seamos Preguntones", se preguntaba cómo se podía saber la dirección inicial y final de un programa utilizando el HESMON, y la respuesta, por supuesto era la correcta, pero existe otra posibilidad de saberlo sin necesidad de cargar el programa en la memoria y es la siguiente. Se hace:

←#NOMBRE PROGRAMA

(NO hay que poner comillas al NOMBRE PROGRAMA)

Y por supuesto en la unidad de disco debe estar el disco que contenga el programa NOMBRE PROGRAMA.

Al escribir lo anterior y pulsar RETURN, la unidad de disco comenzará a girar y al poco tiempo aparecerá en la pantalla 2 números, que por supuesto son las direcciones de inicio y final del programa NOMBRE PROGRAMA.

Esta instrucción no carga en programa en memoria del ordenador, es decir, si deseamos cargarlo debemos utilizar el método normal L"NOMBRE PROGRAMA".

Sin más por esta carta, sólo felicitaros por la estupenda revista que mensualmente publicáis, se despide de vosotros vuestro amigo:

José Antonio Fernández Villoria
C/ García, 11. 2-Izq.
Gijón (Asturias)

DEMOSTRACION DEL SUPEREXPANDER

A la vista de las pocas colaboraciones que se ven en la revista sobre la aplicación del SUPEREXPANDER me he decidido a mandarles este corto programa que pone de manifiesto las magníficas propiedades del cartucho:

```
10 POKE36879,8
20 GRAPHIC 2
30 REGION 7
40 FOR T = 1 TO 100 :STEP2 : G=G+10
50 Y=SIN (T/16)*100+512
60 CIRCLE2,512,Y,G*2,G
70 NEXTT
```

(Pinta la función seno tridimensionalmente).

Puede ser interesante la experimentación con el programa, intentando dibujar otras funciones o mejorándolo gráficamente. Aquí tienen dos ejemplos de lo que se puede hacer con él:

—Cambio de función: se puede cambiar SIN de la línea 50 por COS.

—Efectos gráficos: añadir:

```
60 CIRCLE2,512 ,Y,G*2,G,35,15
65 CIRCLE2,512,Y+100,G*2,G,15,35
```

Espero que mi programa haya servido para que los lectores intenten otras aplicaciones del SUPEREXPANDER y así se aumenten las colaboraciones sobre este tema.

José C. Sarria Moscoso
C/ Barcenillas, 4-4º O
29012 Málaga

BORRADO PARCIAL DE PROGRAMA

Si deseamos borrar la **parte posterior** de un programa y no queremos ir línea por línea, se puede efectuar de la siguiente forma:

1) Detrás de la línea última que queramos mantener, creamos una línea cuyo texto sea ::: (o sea, tres dobles puntos). Ej. 150 :::

2) Al final del programa creamos las líneas siguientes:

```
65500 FOR T = PEEK(44)*256 TO 9↑9:IF PEEK(T-2)+PEEK
      (T-1)+PEEK(T)< >3*58 THEN NEXT
65510 POKE T-6,0:POKE T-5,0
```

Para que funcione hacer RUN 65500

Nota: Sirve para C-64 y VIC-20. La línea de los tres puntos dobles es la señalizadora para el comienzo del borrado.

Federico Iborra
Suscriptor nº 1376

TRUCOS PARA EL NUEVO 16

Soy un nuevo socio de vuestra revista. Poseo un C-16 y os mando una colaboración para la sección de Magia; consiste en:

—Pulsar simultáneamente RUN-STOP y RESET para visualizar el monitor de código máquina.

—A continuación pulsar M y RETURN. Aparecerá en pantalla algo extraño, que supongo serán las primeras direcciones de memoria con sus valores correspondientes. En el margen derecho veréis una banda en RVS-ON que contiene una serie de símbolos.

—El siguiente paso consiste en posicionar el cursor al principio de la cuarta de estas nuevas líneas, concretamente sobre el signo "mayor". Pulsar de nuevo M-RETURN y aparecerá un listado de la memoria RON (creo) que pasará lentamente pulsando la tecla COMMODORE. Si tenéis paciencia observaréis en la franja oscura todos los comandos BASIC, los mensajes de error, los del datassette y disco,...

Víctor José Gallardo Martín
C/ Moreras, 4 Bajo
10003 Cáceres

GANANDO VELOCIDAD EN NUESTROS PROGRAMAS

Les envío unas colaboraciones para la sección de Magia, espero que sean de su agrado.

Para ganar velocidad en los programas en Basic:

—Numerar el programa con números consecutivos (10,11,12,...) y cuanto más bajos mejor.

—En los bucles for...next, en el next poner simplemente NEXT, sin variable.

—Si es posible, utilizar variables sin matrices ni arrays.

—Si en un programa hay sentencias DATA, ponerlas al principio del programa.

—Al hacer una variable a cero poner X=. ,

En cambio, si quieres ver cómo trabaja el VIC-20 secuencia a secuencia pon POKE 37879,0 , (a medida que aumentes el número verás cómo va ganando un poco de velocidad).

Si quieres retrasar la velocidad de movimiento del cursor pon POKE 37879,255.

Juan Clapes Ribas
C/ Madrid, 35-5-2.ª
Telf.: (971) 310992
Ibiza (Baleares)

CONTRASEÑA

En alguna ocasión habréis querido que en un programa hubiera que teclearse una CONTRASEÑA SECRETA para tener acceso al mismo.

Este corto programa permite teclear la contraseña pero sin que se vea lo tecleado en la pantalla, así si hay alguien cerca de nosotros y si somos rápidos tecleándola no sabrán cuál es la palabra secreta.

Está hecha con GET, tantas como letras tenga la contraseña inventada.

Además podéis añadir un campo contador y un condicionante y si se teclea la contraseña varias veces incorrectamente podéis llevarlo a una subrutina de despedida que indique algún mensaje y que acabe con SYS 64738.

Antes de la sentencia SYS 64738 hacer un FOR/NEXT que retarde el borrado del programa y mantenga el mensaje un rato.

Tanto POKE 788,52 como POKE 774,255 son efectivos una vez corrido el programa.

Si se carga el programa mediante LOAD se puede listar el programa una vez cargado, cosa que sería fatal ya que se sabría el nombre de la contraseña.

Para evitar eso es preferible salvar el programa sin nombre y cargarlo mediante SHIFT y RUN/STOP para que el programa una vez cargado corra sin necesidad de teclear RUN.

```
  1 REM CONTRASEÑA INVISIBLE
  2 POKE 788,52 : REM BLOQUEA TECLA RUN/STOP
    UNA VEZ CORRIDO PROGRAMA
  3 POKE 774,255 : REM IMPIDE LISTAR UNA VEZ
    CORRIDO EL PROGRAMA.
  4 Z$ = "SEBAS" : REM INDICAR NOMBRE DE
    CONTRASEÑA.
  5 PRINT"CLR/HOME"
 10 PRINT"TECLEE CONTRASEÑA"
 20 GET A$ : IF A$ = "" THEN 20
 30 GET B$ : IF B$ = "" THEN 30
 40 GET C$ : IF C$ = "" THEN 40
 50 GET D$ : IF D$ = "" THEN 50
 60 GET E$ : IF E$ = "" THEN 60
 70 REM COLOCAR TANTAS GET COMO LETRAS
    TENGA LA CONTRASEÑA
 80 IF A$+B$+C$+D$+E$<>Z$ THEN GOSUB 5000
 90 IF A$+B$+C$+D$+E$ = Z$ THEN 110
100 GOTO 5
110 FOR X = 1 TO 2000 : NEXT X : PRINT "CLR/HOME"
120 PRINT TAB(5) "HOLA AMIGO" ; Z$
130 PRINT TAB(3) "ESTOY LISTO PARA TRABAJAR"
140 FOR X = 1 TO 4000 : NEXT X
```



```
4500 END : REM ANTES DE END TECLEAR EL PROGRAMA
5000 PRINT TAB(5)"CONTRASEÑA NO VALIDA"
50001 FORX = 1 TO 3000 : NEXT X : RETURN.
```

Sebastián Soler Alonso
Verdi, 44-3º
Telf.: 237 69 47
08012 Barcelona.

ERRORES EN EL INPUT

"Cuando hacemos un INPUT con una variable numérica (por ejemplo INPUT A) corremos el riesgo de que el operador introduzca por equivocación una letra con lo que saldrá en pantalla un mensaje de error que puede estropear una presentación elaborada. Esto se puede arreglar en gran parte con este truco:

```
1 Input a$:a=val(a$);If a=0 then 1
```

Esto mismo se puede hacer con GET. La variable resultante es una variable numérica(a) siempre que no sea una letra. En tal caso vuelve a hacer el input sin que aparezca ningún mensaje de error.

René Suárez Heria
Tirana-Lariana (Asturias)

PANTALLA MOVIL EN EL C-64

En primer lugar, deseo felicitaros por vuestra revista, pues con ella he aprendido mucho. Bueno, ahí va lo que os tengo que decir: son estos POKEs:

POKE 53265,87 (Se encoge la pantalla y el cursor aparece de color rojo, pero al escribir, escribe en color que se quiera).

Si se van disminuyendo los números (86,85...), hasta 80, las letras van subiendo.

Carlos Acuña Quiroga
Carrera del Conde, 2-4º C
Santiago de Compostela (La Coruña)
Galicia

"TRON" MEJORADO

Muy satisfecho por el viraje a programos en C/M que está dando definitivamente "nuestra" revista, os mando una mini-contribución conectada con TRON de J. M. Beltrán, publicada en el nº 14, pág. 54.

Yo estaba tan cansado de ver cómo la ventanita del trazador del Simon's Basic era algo así como esas persianas con resorte, que suben como una bala, y te deja a dos velas, que iba a fabricarme un trazador. Pero como el único defecto del TRON este es, que el "freno" con un lazo es poca cosa y la cosa va que marea, ahí os mando un trinquete, tal que con mínimas modificaciones el TRON va al paso de pulsar la tecla del logotipo y luego su vecina Shift. Si no pulsas nada más, te puedes ir a tomar café, que la cosa no pasa del sitio.

Los cambios obvios, son:

```
$C066 LDA $028D   } Espera que pulses Shift,
      CMP #$01    } después de pulsar logotipo.
      BNE $C066
      NOP
      NOP
       :        :
 C06F NOP
 C070 Idem TRON del amigo BELTRAN
```

José M. Paneque Rodríguez
Avda. Andalucía, Bl. 2, 3º C
Cabra (Córdoba)

MARKETCLUB

Servicio gratuito para nuestros lectores *PARTICULARES*. Los anuncios serán publicados durante 1 mes. Debido a ciertos abusos que nos han sido comunicados, los anuncios gratuitos de Market Club *SOLAMENTE* serán publicados si vienen con nombre, apellidos y dirección completa.

MERCADILLO

• Poseo un Commodore 64, datassette e impresora MPS-801, así como una amplia biblioteca de programas de todo tipo. Me gustaría ponerme en contacto con usuarios para intercambio de programas. Me interesan, en especial, utilidades para impresora. Luis Luna Barrios. P/ Gomila, 16 - 10. Tel. (954) 51 33 45. 41007 Sevilla. ***(Ref. M-383).***

• Intercambio programas —en cinta— para el VIC-20 de juegos y utilidades, con o sin ampliación de memoria. Escribidme a: Francisco Jesús Gutiérrez Izquierdo. C/ Prolongación de Escañuela, 2. 14002 Córdoba. ***(Ref. M-384).***

• Me gustaría contactar con usuarios del VIC-20 para intercambio de programas y experiencias sin fines lucrativos. Llamar o escribir a José Tomás Gómez Valencia. C/ María Díaz de Haro, 163. Tel. (94) 441 02 35 (noches). 48013 Bilbao. ***(Ref. M-385).***

• Vendo para el VIC cartucho de juego "Jupiter Lander" o cambiaría por juegos. Desearía entablar correspondencia con Vic-ciosos para intercambio de juegos, compraría joystick o cambiaría. Dirección: Javier Fernández Iglesias. C/ Ramiranes, 4. Parriños (Pontevedra). ***(Ref. M-386).***

• Cambio controlador doméstico de 8 relés nuevo para VIC-20 o CBM-64 con manuales por unidad de cassette Commodore. También cambio cartucho para VIC-20 "ROAD RACE" por joystick Commodore. Tomás Gómez Valencia. C/ María Díaz de Haro, 16 3F. 48013 Bilbao. ***(Ref. M-387).***

• Vendo Commodore 64, con cassette, joystick y juegos. Además del manual de la guía de referencia C-64. Todo por 55.000 ptas. También muchas revistas. Sergio Vernis Perraman. C/ Mavern, 11 - 5º 1. Barcelona. Tel. 209 39 64. ***(Ref. M-388).***

• Vendo VIC-20 con superexpander 3K, cassette CN2, funda, joystick, curso Basic I y II parte, cartucho de juegos, muchos programas en cinta, revistas y libros. Todo en perfecto estado de uso por 40.000 ptas. Juan Giménez Franco. C/Daoíz y Velarde, 24 -2º. 08028 Barcelona. Tel. 330 51 48. ***(Ref. M-389).***

• Vendo ordenador VIC-20 con ampliación de 16K, mucha información y programas; además programas de juegos y programa de 40 columnas. Todo 39.000 ptas. Antonio López Sánchez. C/ Salado, 3 - 2º D. Cartagena (Murcia) (968) 53 60 27. ***(Ref. M-390).***

• Vendo o cambio por impresora para Commodore-64 cantidad de programas de utilidades y juegos para el C-64. Pedro Cazorla López. Argentina, 7. Telf. 69 10 93. Telde - Las Palmas. ***(Ref. M-391).***

• Compro controlador de 8 relés, o cambiaría por cartucho de juego International Soccer (fútbol) y programas en cinta Organo y B1 Bombardero Nuclear (**son originales, no copias**). Juan Caballero Díaz. Bº Gaudí, Bl. C 3.ª nº 8. Reus (Tarragona). ***(Ref. M-392).***

• Vendo o cambio ordenador CBM VIC-20 en perfecto estado de funcionamiento, junto con la grabadora CN2, un joy de mando anatómico, cartuchos SRGON II y Radar-Strace, Manual del Usuario, Guía de Referencia, Primera parte del curso de introducción al Basic en dos cintas, libros y revistas especializadas y programas en cintas. Todo junto o por separado, interesados escribir a la siguiente dirección: Av. de Madrid, 203 B 3º 3.ª, 08014 de Barcelona. Todo junto está valorado en 60.000 ptas., precio a convenir. Desearía intercambiar información sobre programas de aplicación para estadísticas de baloncesto, (no importa el tipo o marca de ordenador). José de Francisco. Av. de Madrid, 203 B 3.ª, 3.ª. 08014-Barcelona. ***(Ref. M-393).***

• Vendo ordenador Commodore VIC-20 del año 84, completo más programas y cartucho, todo por 20.000 ptas. Dirigirse a Julián Sánchez Alonso. c/ Tunaima, 2-5º D. 28033-Madrid. Telf.: 763 49 63. ***(Ref. M-394).***

• Urge vender cuanto antes VIC-20 + cartucho SUPEREXPANDER + cartucho de juegos Road Race + libros introducción al lenguaje de programación Basic I y II (con sus cintas correspondientes) + manual del usuario (español e inglés). También regalaría 2 cintas de 30 juegos en código máquina con su lista de exploración de los programas para realizar intercambios. Todo ello por 29.000 ptas. Escribid o llamad a Carlos Gustavo c/ Callado, 7. 4º D. 41013-Sevilla. Telf. (954) 61 41 24. Mejor de 1,30 a 3 h. o de 19,30 a 22 h. ***(Ref. M-395).***

• Urge vender VIC-20 (abril 84) en perfecto estado (con embalaje de la casa), con manual en castellano y en inglés, libro "Acceso rápido al VIC-20" por Tim Hartnell, 3 cintas con programas y muchos listados de programas para VIC-20. Lo vendería todo por unas 22.000 ptas. Interesados llamar de 2 a 3,30 ó de 9,40 a 10,30 preguntando por José, mi dirección es: José Francisco Alonso. c/ Cruz y Castillo, 8-5º. 34004 Palencia. Telf.: (988) 72 80 99. ***(Ref. M-396).***

• Vendo VIC-20, 1.ª y 2.ª parte del curso de Basic, ampliación de memorias 3,5 K y 16 K, juegos de la casa INDESCOMP, Traganúmeros, Trenzador, Myriads, Frogger, Asteroides, Blikzkrig, Comecocos, etc., etc....II Chess y muchos más. Sargon. Se puede vender todo por separado o bien por lotes, según interese, el precio no habrá problema. **También pedir que si alguien me pudiera facilitar el programa para el Commodore 64 de CW y RTTY para el interfase VIG-HAM).** Interesados ponerse en contacto con Ramón Petit a los telfs.: 336 26 39 o bien al 335 45 46. También pueden escribir a mi dirección: c/ Freixa Llarga, s/n. L'Hospitalet o también el apartado 2020 de L'Hospitalet de Barcelona. ***(Ref. M-297).***

• Por cambio de ordenador, vendo VIC-20 nuevo (adquirido en dic.-84) en perfecto estado, con los libros de acceso rápido al VIC-20 (más de 50 programas) y guía del usuario, un cassette original de juegos y otro con algunos programas, todo por 20.000 ptas. Llamad o escribid a Carlos González. c/ Jacinto Verdaguer, 21. Amposta. Tarragona. Telf.: (977) 70 14 15. ***(Ref. M-398).***

• Soy profesor de química y física. Me interesaría instrucciones lenguaje Logo, Forth y Pascal para el Commodore 64, y en general programas de aplicaciones, particularmente sobre mi materia. Escribid o llamad a Enric Buchaco. C/ Andrea Doria, s/n. I.B. Salvat Papasseit. 08003-Barcelona. Telf.: 219 87 46 ó 315 02 03. ***(Ref. M-399).***

• Vendo VIC-20 del 84, 20.000 ptas.; Dattasete C2N (casi nuevo), 8.500 ptas.; cinta Graphics, 1.000 ptas.; cinta Skramble, 800 ptas.; cinta nigtmare Park, 400 ptas. Todo junto, 29.000 ptas. Llamar de 2 a 3 del mediodía, preguntar por José. José Tuches Boveda. c/ Alonso Carbonell, 2. 28045-Madrid. ***(Ref. M-400).***

• Vendo VIC-20 y televisión B/N 12" marca Inter, listo para funcionar, con garantía desde enero-85, dos manuales, uno en inglés y el otro en español, un libro de introducción a los ordenadores por 35.000 ptas. al contado. Con instrucciones en español para su uso. Llamar sólo viernes de 3 tarde a 9 noche. Nemesio Solís García. c/ Pujadas, 265 ático. 08005-Barcelona. Telf.: 303 32 40. ***(Ref. M-401).***

• Vendo VIC-20 en perfecto estado por 20.000 ptas + Guía del usuario del Vic (1.500 ptas.), todo junto por 22.500 ptas. también por separado. Regalo libro "acceso rápido al Vic-20" y algunos juegos. Manuel Aranda. Río Sella, 10. Móstoles. Madrid. ***(Ref. M-402).***

• Vendo el cartucho y disco de instrucciones de contabilidad 64, 300 cuentas, 3.000 ajustes, también vendo el cartucho del programa Magic Disk por juntos o separados, precio a convenir. Agustín Rosco Vázquez. Plaza de la Merced, 11. Azuaga. Badajoz. Telf.: 89 03 56. ***(Ref. M-403).***

• Vendo VIC-20, ampliación de 16K y tarjeta 40/80 caracteres por 35.000 ptas. Regalo juegos y programas. Antonio López Sánchez. Salado, 3. 2º D. Cartagena. Murcia. ***(Ref. M-404).***

• Desearía comprar Superexpander + 3K para un VIC-20. Precio razonable. Los interesados llamar los jueves de 19 a 20 o escribir a Luis Campoy Castro. c/ Ugarte Dña. María, 14 A 3º Tudela Navarra. Telf.: 82 26 15. ***(Ref. M-405).***

• Vendo cartucho Easy Calc, Magia Desk y Music machine o cambio por monitor FV o N. Valor de compra 40.000 ptas. y doy por 20.000. También vendo unidad de disco e impresora Seikosa en 70.000 ptas. todo en perfecto estado y con instrucciones en español. Contactar con Jesús Benito. Telf.: 248 00 38. Madrid. ***(Ref. M-406).***

• Vendo VIC-20 + ampliación 3K + 40 programas en cinta (Juegos y aplicaciones) + 3 libros. Todo 25.000 ptas. José Salgado Herrera. c/ Donoso Cortés, 58 1º D. 28005-Madrid. ***(Ref. M-407).***

• ¡Atención usuarios VIC! Vendo cassette C2N por 9.000 ptas. Perfecto estado. Ramón García Atance. Madrid. Telf.: 419 66 59. ***(Ref. M-408).***

• Urge vender VIC-20 por 19.500 ptas. (prácticamente nuevo) + cassette C2N por 8.500 ptas. + 1 cartucho (sargon II chess) por 2.000 ptas. + cinta indescomp (Shadowfax) por 950 ptas. y además regalo: 15 revistas Commodore + cintas con juegos + curso iniciación al Basic. Roger Llobet García. c/ Numancia, 115-7º 1.ª. 08029-Barcelona. Telf.: (93) 239 81 42. ***(Ref. M-409).***

• Vendo VIC-20 en perfecto estado (casi nuevo) con garantía, Datasette (nueva), unos 90 juegos comerciales (en 4 cintas), libros "Manual del usuario" y "Introducción al Basic: parte 1", un libro para aprender Basic, un paquete de diskettes con juegos y 2 joysticks: -Quick shot 1, usado unas veces; -Quick shot 2, totalmente nuevo. Precio a convenir. Dirigirse a: Daniel Garriga, C.N.2, c/ Walden, 7. Sant Just Desvern. Barcelona. Telf.: 372 02 43. ***(Ref. M-410).***

• Desearía comprar ampliación de memoria de 16 Kb. para el VIC-20, a ser posible con la condición de un período de prueba de unos días. Seriedad absoluta en el trato. Jesús Cea Méndez. c/ Caldas de Reyes, bloque D-61, portal 2-6º izq. Telf.: (986) 23 18 35. (Llamar después de las 15 horas). Vigo-9 (Pontevedra). ***(Ref. M-411).***

• Cambo Commodore 64 por Dattassete por otro ordenador de similares características (preferible MSX) o vendo ambas cosas por 59.000 ptas. Con garantía, conexiones, adaptador y manual. Envío rápido y sin gastos de envío. Francisco Rocha Betancor. c/ Carretera del Centro, 7. 35017-Las Palmas. Telf.: (928) 35 35 50. ***(Ref. M-412).***

• Vendo ordenador VIC-20 más cartucho ampliación de memoria de 8 Kbytes con toda la correspondiente instalación. Todo como nuevo. Precio inicial 25.000 ptas., a negociar. Regalo de guía de referencia del programador VIC-20. Llamar al teléfono (93) 371 18 15. Miguel Sanchís Flores. Calle Norte, 63. 3º 2.ª. Sant Just Desvern. Barcelona. ***(Ref. M-413).***

• Vendo consola de vídeo juegos ATARI, con dos joysticks y 7 cintas, por 17.000 ptas., nuevo. Precio real: 30.000 ptas. Llamar a Luis Figueras. Telf. (93) 253 51 19. c/ Consejo de Ciento, 170. 8º B. ***(Ref. M-414).***

Deseo contactar con otros amigos Commodorianos

Nombre ....

Dirección...

Telf.: .... **Ciudad:**......................

C.P. .... **Provincia** ....

Modelo de ordenador..

Tengo Cassette .... □

Unidad de Disco... □

• Compraría cassette, libros y accesorios para el C-64. También estaría dispuesto a intercambiar programas. Interesados llamar al teléfono (958) 11 98 20 o escribir a José Luis Gallardo García. c/ Circunvalación, nº 2. 1º C. 18007-Granada. *(Ref. M-415).*

• Vendo VIC-20 por 15.000 ptas. Ampliación de 16K con interruptores por 10.000 ptas. Ampliación hasta 35K con 8K Eprom para grabar por 18.000 ptas. Complementos (libros, documentación y 300 programas), por 15.000 ptas. Javier. Telf.: 218 96 59. c/ Teruel, 4. Colonia de Los Angeles. 28023-Madrid. *(Ref. M-416).*

• Vendo impresora Commodore MPS 801, comprada en enero de 1985. Regalo al comprador programas para su C-64. Precio a convenir. Deseo además contactar con usuarios de C-64 de toda España para intercambio de programas. Víctor Calvo Medina. c/ Beatas, 20. 3-D. 29008 Málaga. Telf.: (952) 21 44 09. *(Ref. M-417).*

• Vendo CBM-64 con Datassete, cintas de juego y revistas por 60.000 ptas. Eduardo Sánchez Muñóz. Jaraíz de la Vera, 12-3 izq. 28011-Madrid. Telf.: (91) 470 01 66. *(Ref. M-418).*

• Compro Commodore-64 o lo cambio por Spectrum Plus 64K más novecientos programas; ofertas J.M. Martín Santos. c/ Moncado, 10-2º, 4º. 12005 Castellón. *(Ref. M-419).*

• Vendo para VIC-20, expansor y cartuchos y alojamiento Eprom (5.000 ptas.). Superexpander + 3K (5.000 ptas.). Cartuchos: Sargon II Chess (2.500 ptas.); Avenger (1.000 ptas.); Mole Attack (1.000 ptas.), Commodore Magazine (núms. 1 a 12) (1.500 ptas.). Commodore World (núms. 1 a 14 + Especial 100 programas) (2.500 ptas.). Club Commodore (núms. 1 a 15) (1.000 ptas.). Guía referencia del programador (1.000 ptas.). Guillermo Crovetto Montoya. c/ Pedro Antonio de Alarcón, 62, 1º A. 18002-Granada. Telf.: (958) 28 03 76. *(Ref. M-420).*

• Vendo cartuchos Forth y Simon's Basic (con instrucciones) para Commodore 64 en perfecto estado. Precio a convenir. Dirigirse a: Eusebio Zuloaga. Telf.: (91) 247 22 76. (Tardes). *(Ref. M-421).*

• Vendo ordenador DRAGON-2 por estrenar, manuales en castellano, con programas (juegos, técnicos, educativos). precio a convenir. Dirigirse a: Eusebio Zuloaga. Telf.: (91) 247 22 76. (Tardes). *(Ref. M-422).*

• Vendo Commodore VIC-20 + Datassette Modelo C2N + Cartucho juego "AVENGER" + Cinta juego "Skramble" + Programa base de datos + Introducción al lenguaje de programación Basic del valor VIC-20: 1.ª parte + Manual del VIC-20; todo ello adquirido entre noviembre del 84 y febrero del 85, por sólo 30.000 ptas. El ordenador lo vendería por 28.500 ptas. Si alguien está interesado, escribir a esta dirección: Carlos Galán. c/ Mestre Millet, 6, bajos 2.ª. Gavá (Barcelona) o llamen al teléfono (93) 662 50 90 entre las 1,45 y las 2,20 de la tarde, pregunten por Carlos. *(Ref. M-423).*

• Vendo impresora Commodore VIC-1515 muy nueva, precio 30.000 ptas. También vendo programa contabilidad de 300 ctas. consta de cartucho, disco e instrucciones por 17.000 ptas. Además, intercambio experiencias y programas para el C-64. Luis Colino Asensio. Avda. de Colón 12-3D. 06005-Badajoz. *(Ref. M-424).*

• Vendo VIC-20 con el primer curso de Basic, cartucho de ajedrez y libro de ayuda al programador, todo por 25.000 ptas. Acepto otras ofertas, llamar a Francisco Bazán Solano. Telf.: (93) 217 22 35. c/ Santa Perpetua, 17 3º 1.ª. 08012-Barcelona. *(Ref. M-425).*

• Regalo por 15.000 ptas. un VIC-20 en perfecto estado junto con manuales, curso de Basic, listados de programas, y todos los programas de que dispongo. Miguel A. Puerta Carrasco. c/ San Jerónimo, 23, 2º Izq. 18001-Granada. Telf.: 27 69 78. *(Ref. M-426).*

• Vendo VIC-20. Casi nuevo y en muy buen estado. Incluyo: Manuel del usuario, Guía de referencia, 1.ª parte de curso de introducción al Basic (con dos cintas) y varias cintas con más de 200 programas de todas clases. Todo por 25.000 ptas. Precio real aproximado: 45.000 ptas. Lunes a sábado a partir de 16 h. (971) 34 17 64. José M.ª Sánchez González. c/ Del Mar, 18. 1º-3.ª. San Antonio Abad - Ibiza - Baleares. *(Ref. M-427).*

• Vendo, por no serme de utilidad, programa Master para el C-64, compuesto por diskette, llave electrónica y manual en castellano y en inglés, totalmente a estrenar. También aceptaría combinarlo por algún otro programa de los denominados "hojas de trabajo" como Calc Result o Practicalc, etc. Santiago Ovejero. c/ Tembleque, 124. 28024-Madrid. Telf.: 711 97 77. *(Ref. M-428).*

• ¡¡¡Vendo VIC-20 nuevo, comprado en agosto de 1984, más: El manual del usuario, Introducción al lenguaje de la programación Basic, parte I, más de 30 juegos. Todo en perfecto estado. A 24.000 ptas. Mi dirección es: Pablo Lestau Sáenz, Apdo. 68. Calahorra (La Rioja). Telef.: (941) 131 17 61!!! *(Ref. M-429).*

• Vendo VIC-20 (abril 84), ampliación de memoria de 16K (diciembre 84), guía de referencia del programador y dos programas originales de Indescomp. Interesados llamen al teléfono (957) 25 08 73 (preguntar por Francisco Jesús) de 19 h. a 21 h. o escriban a las siguientes señas: Francisco Jesús Gutiérrez Izquierdo. c/ Prolongación de Escañuela nº 2. 14002-Córdoba. *(Ref. M-430).*

• Vendo cartucho Simon's Basic, con solo cuatro horas de uso, por tenerlo repetido. También lo cambiaría por material para C-64. Mandar número de teléfono si tenéis prisa. Luis Tudela Casanovas. Pasage de la Tossa, 20 (entr.). 08012-Igualada (Barna). *(Ref. M-431).*

• Vendo VIC-20 + grabadora + ampliación de 16K y 3K + cassettes con más de 60 juegos + muchas revistas + guía del usuario en inglés y en castellano + conexión para Tv blanco/negro, todo por 30.000 ptas. La grabadora está en garantía de "El Corte Inglés". Comprado todo menos la grabadora en agosto 84. La grabadora a finales de enero 85. Dirigirse a Enrique. Tel. (93) 209 56 07. Vía Augusta, 195. 08021-Barcelona. *(432).*

• Urge vender Spectrum de 48K (comprado en nov. del 84), Joystick con su correspondiente Interface tipo Kempston (comprado en febrero del 85) y un cassette especial para ordenador (comprado en diciembre del 84). Junto con los mejores 70 juegos para el Spectrum y algunas revistas de informática. Llamar al (91) 416 72 20 o bien escribir a José Clemente González. c/ Corazón de María, 27. 7º E, 28002-Madrid. Todo está en un estado impecable, ya que apenas ha sido usado. El precio total es de 49.000 ptas. (se puede discutir). *(Ref. M4-33).*

• Cambio los 12 conocidos programas comerciales por los números 1, 2, 3 de Commodore World que están en buen estado. Los programas están grabados en FAST-TURBO. Incluyo instrucciones detalladas de los programas. Son todos estupendos. *(Ref. M-434).*

• Cambio y vendo programas para VIC-20. Daniel Roig Marchuet. c/ Castilla, nº 27, bajos. Ibiza (Baleares). *(Ref. M-435).*

CLUBS

• Club de amigos. Foto Estudio 2. Plaza de Sombrereros, 2. **Palma de Mallorca**. Tel.: 21 31 62. *(Ref. C-1).*

• Club de programación Alaiz. Tels.: 254 480-257 704. Pamplona. *(Ref. C-4)*

• Club de Commodore de Albacete. Fernando Martínez Guerrero. La Roda, 39, 5º D. Albacete. *(Ref. C-5)*

• Círculo de Durensan Vigués. Vigo. C/Venezuela, 48. Entrechan. Tels.: (986) 410 683/422 519. *(Ref. C-6)*

• Asociación Manchega de Usuarios de CBM-64 (A.M.U.C.-64). c/ Blasco de Garay, 10. Albacete. c/ La Roda, 39-5º D. c/ Octavio Cuartero, 15-6º D. *(Ref. C-13).*

• Interesados en la formación de un club de informática en Granada capital, diríjanse por correo a c/ San Jerónimo, 23-2º, 18001 Granada. Miguel A. Puerta Carrasco. *(Ref. C-28).*

• Para formación club nacional de CBM Series 2000 y 3000 exclusivamente, ponerse en contacto por carta con Artemio González Pérez. Fernando Arocena Quintero, 1. Teléf.: 22 42 44. 38009 Santa Cruz de Tenerife. *(Ref. C-31).*

• Ha nacido el club "Brian & Chip" para usuarios del C-64 y Spectrum 16K. Crearemos una revista propia. Escribir desde cualquier rincón de España. Francisco Reig López, C/Marina, 261, 3º 5.ª, 08025 Barcelona. *(Ref. C-33).*

• Desearía información sobre clubs en existencia y gente que quiera formar uno en Pontevedra de Commodore 64. Carlos Parada Gandos. C/Padre Fernando Olmedo, 4-4º B. 36002 Pontevedra. Tel. (986) 85 69 21. *(Ref. C-43).*

• Ha nacido el Club de Informática y videojuegos, el que nos escriba recibirá su carnet, con nº, intentaremos crear un presidente por ciudad (el 1º en escribir de cada una). Juan A. Manso. C/Illescas, 81. 28024 Madrid *(Ref. C-44)*

• Zaragoza. Club de amigos del VIC-20 y C-64 en ESPUMOSOS los sábados de 9,30 a 12,30. Daniel Fdez. de Velasco, C/Gran Vía, 39. Zaragoza. *(Ref. C-45).*

• Les ruego si puede ser que me pongan en contacto con algún usuario del VIC-20. Daniel Roig. C/Castilla, 27. Ibiza (Baleares) *(Ref. C-46).*

• Me gustaría contactar con alguien de **Figueras** o alrededores para intercambiar programas o formar un Club Commodore 64. Josep Teixidor Planas, Vilasacra, 2. El Far D, Emporda (Girona). Tel. 508649. *(Ref. C-47).*

• VIC-20 Desearía un intercambio de experiencias en cuanto a programación, aplicaciones al campo de la radioafición (intefaces) y programas diversos. EC 6 MF. Lorenzo Sabater Coll. C/Agua, 2. Esporlas, Mallorca, Baleares. *(Ref. C-48).*

• Estamos formando un club de usuarios del COMMODORE 64 en Las Palmas. Desearíamos contactar con chicos y chicas para intercambio de programas y experiencias. Interesados dirigirse a: César Montenegro Armas, C/Pío XII, 56 2-D. Tel. 24 60 61. *(Ref. C-49).*

• Desearía ponerme en contacto con usuarios del VIC-20 para intercambiar programas e información. Carles Balañá. C/Tívoli, nº 11, 5ºA. Réus (Tarragona). *(Ref. C-50).*

• Som dos nois de SABADELL que estem interassats en formar un club d'usuaris d'ordenadors COMMODORE i SINCLAIR. Els interassats o interassades, poden escriure a: Daniel Ribes Garolera, C/Via Aurelia nº 24, 2º 3º Sabadell. Tambe demanen si es posible informació sobre clubs ja existens. *(Ref. C-51).*

• Desearía contactar por correo con usuarios del VIC-20, tanto usuarios nacionales como extranjeros. Escribir a Mª Mercedes a la C/Marqués del Muni, 25 (Guía de Gran Canaria). *(Ref. C-52).*

• Se ha creado "COESPECTRUM 16", un nuevo Club, nacido para todos los usuarios de CBM 64, Spectrum 16K y Dragon 32 y 64. Pero para que todo ello se haga realidad, necesitamos colaboradores y muchas ideas. Se intercambiarán toda clase de programas o listados. Se creará una revista. Escribir a Avd. de la Aurora, 57, 12º A. (29006) Málaga, desde cualquier punto de España, se contestará o llamar a: Angel, Tel. 321799, o Emilio, Tel. 398169. *(Ref. C-53).*

• Desearía contactar con interesados para formar un club de usuarios en Vigo (Pontevedra) (o sus alrededores). A ser posible entre 12 y 16 años (aunque de otra edad me es prácticamente igual). Telf. 37 86 77. Preguntar por José Manuel Mera Alonso. C/ Buenos Aires, 2-2º C. Vigo (Pontevedra). *(Ref. C-54).*

• Club Commodore del Bager con una amplia programoteca (más de 250 programas) y gran cantidad de accesorios para CBM 64 y Vic-20 deseamos contactar con otros clubs y usuarios para intercambio. Dirigirse a: Félix Portabella. C/ Forn de Sta. Llucia, 1-2.ª. Manresa (Barcelona). Telf. 872 72 97 u Oscar 872 20 71. *(Ref. C-55).*

• Me gustaría contactar con usuarios de Commodore 64 de La Coruña para intercambiar programas y experiencia sin que nos cueste nada si se dan las condiciones apropiadas. Si estais interesados mis señas son: Gonzalo Montoto Veira. Avd. Sardiñeira, 5-2º. 15007 La Coruña. *(Ref. C-56).*

• Quisiera contactar con usuarios de Commodore-64, para formar un club aquí en Málaga. Podríamos compartir muchas experiencias, programas, etc. y salir ganando todos. Contactar con Víctor Calvo Medina. Telf. 21 44 09 de Málaga. *(Ref. C-57).*

• Agradecería recibir direcciones de commodorianos-64 residentes en Guipúzcoa en especial Irún, Fuenterrabía, Mondragón y Vergara. Manuel Prieto Ugarte. C/ Izagu, 5. Irún (Guipúzcoa). Telf. 62 47 38. *(Ref. C-58).*

• Me gustaría recibir información sobre clubs en Sevilla, como para poder intercambiar programas, etc. Pablo Jesús Astorga Lara. Residencia Conde de Bustillo, bloque 11-7º F. Telf. 63 52 37. Sevilla-5. *(Ref. C-59).*

• Si deseáis formar un club de usuarios de ordenadores VIC-20 en Córdoba, escribidme a: Francisco Jesús Gutiérrez Izquierdo. c/ Prolongación de Escañuela, 2. 14002-Córdoba. Telf. 25 08 73 (de 19 a 21 h.). *(Ref. C-60).*

• SPRITE. Club de usuarios de Commodore 64. Sin ánimo de lucro. Dirigirse a Pedro Javier Muñóz Pérez. c/ Zarandona, 48. Santiago de la Ribera (Murcia). *(Ref. C-61).*

• Desearía formar un club de amistad para toda España, para intercambiar programas para VIC-20, cambiar experiencias, sin gastarnos dinero. Daniel Roig Marchuet. c/ Castilla, 27 (Bajos). Ibiza (Baleares). *(Ref. C-62).*

• Deseo contactar con usuarios de Superbase-64

para intercambio de experiencias y/o programas. Josep Rovira i Sardá. c/ Dr. Escayola, 17-1º-2.ª. Telf. (93) 891 07 40. San Sadurni de Noya. *(Ref. C-63).*

• Desearía contactar con algún club de usuarios de C-64 en Málaga. Marcos García. Rueda. Telf. 27 11 36 *(Ref. C-64).*

• Quiero contactar con chico-chicas que empiezan a programar en código máquina. Intercambiaríamos experiencias, información, etc. Escribir a Robert Martin. Font, 21. Cassa de la Selva (Girona). Tengo un C-64. *(Ref. C-65).*

• Propongo crear una red de intercambio por teléfono mediante modem. Si alguien se lo desea autoconstruir, puedo facilitar esquema y conexionado. Interesa conseguir software para intercomunicación con modem. Llamar al (93) 333 50 80. Pedro Margalef. Cardenal Reig, 23 - 2º 1ª. 08028 Barcelona. *(Ref. C-66).*

• Apúntate al club "SPECDORE", para usuarios del Commodore 64 y Spectrum. Respuesta garantizada. Para inscribirse preguntar por Juanvi en el tel. (943) 45 10 09, o bien, escribir a C/ Prim, 37 - 2º 20006 San Sebastián (Guipúzcoa). *(Ref. C-67).*

• Estoy interesado en ponerme en contacto con usuarios del C-64 que posean algún programa de traducción de textos potente, a ser posible en disco, de inglés a español. Asimismo, quisiera conseguir las instrucciones del compilador Pascal 64 versión 3.0. Quien tenga información sobre esto escriba a: Mariano Prado Zorzano. C/ Albia de Castro, 3. 26003 Logroño. Tel. (941) 23 01 54. Estaría dispuesto a comprarlo o cambiarlo por programas, esto a convenir. *(Ref. C-68).*

• Estoy interesado en intercambiar programas, en especial educativos. También quiero entrar en contacto con profesores de EGB, BUP y FP que estén interesados en la programación, sobre todo si están llevando alguna experiencia en la escuela. Manuel del Arbol Sánchez. C/ Arrumbadores, 4 - 4º. Montilla (Córdoba). *(Ref. C-69).*

• Se ha formado el "SINCOM" Club de Spectrum y Commodore 64. Se creará una revista. Manda sobre con tu dirección debidamente franqueado para contestarte con la máxima rapidez. Escribid a SINCOM. Avda. Virgen de Guadalupe, 20 - 6 F. 10001 Cáceres. *(Ref. C-70).*

• Me gustaría poder contactar con usuarios de Commodore 64 que residan principalmente en Cádiz, aunque también pueden ser de cualquier otro lugar de España. Manuel Jesús Sánchez Mendoza. c/ Stº Domingo de la Calzada, 9-1 Dcha. Telf.: 27 88 44. Cádiz. *(Ref. C-71).*

• En Granada hay un club para toda España, para unir, proteger, premiar la creatividad, confrontar su participación en campeonatos de los mejores juegos existentes y llevar a efecto cuantas ideas redunde en beneficio de los Commodorianos que tienen un modelo 64. Si deseáis amplia información, sólo tenéis que dirigiros a SUPER CLUB-64. Apartado de Correos 456. c/ Joaquín Costa, 6-6º de Granada y rápidamente os remitiremos amplia información. Te esperamos con los brazos abiertos. *(Ref. C-72).*

• Estoy interesado en formar un club de usuarios del VIC-20, Oric atmos y Spectrum. Daniel Roig Marchuet. c/ Castilla, 27, bajos. Ibiza (Baleares). *(Ref. C-73).*

• ¡Atención Commodorianos de la comarca de Lérida! me llamo Antonio y tengo 14 años. Poseo un C-64 y desearía el contacto o comunicación con otros usuarios de C-64 para el intercambio de programas y experiencia (poseo muchos juegos), Soy principiante y podéis escribirme a: c/ Joan Maragall, 96. Preixana (Bellpuig). Lérida, o llamarme al teléfono 32 07 50. *(Ref. C-74).*

DESEO CONTACTOR CON OTROS AMIGOS COMMODORIANOS

C-64

• Jaime Pérez Moreno, Alamos, 5-4º B. Alcalá la Real, Jaén. Telf.: (935) 58 13 38. Sólo tardes a partir de las 3,30 de la tarde. Poseo unidad de disco.

• Jesés Angel Sanz Serrano. c/ Noya, 5-1º C. San José de Valderas. Madrid.

• Freddy M. López González. c/ Puerta Canseco, 47-3º B. 38003 S.C. de Tenerife. Poseo cintas y discos.

• Thomas Kowolik. c/ Sevilla, 21. Castelldefels. Barcelona. Telef.: 664 39 98. Poseo unidad de disco.

• F. Engelberts, Box 1422, 2970 Emden/W. Germany. Poseo unidad de disco.

VIC-20

• Ramón Faura Pedrals. Avinguda II de septembre, 8. Berga (Barcelona). Tel.: 821 05 20.

SUPER-INTERESANTISIMO

PIRATERIA DE SOFT ...Y ALGO MAS

(Viene de pág. 6)

• No me meto en que haya profesionales abusivos y otros que no. Supongo que de todos los colores). Ni cual porcentaje de beneficio es correcto, ni con cuál comienza el abusivo. Lo que sí sé, es que no existe mayor abuso ni pingüe negocio que el del pirata —entre cinta, copiador, tiempo empleado y sello de correos su inversión máxima (dándole superlargas) es de 100 ptas., y eso para redondear—. La publicidad del producto, en los piratas no "profesionalizados" —en la mayoría— la tienen más que hecha por los profesionales. Venden los programas entre 500 y 2.000 ptas. según se trate de un juego o de un programa de utilidades caro.

Si un beneficio de un 400 a un 1.900% no es abusivo... ¡vale!

Y si adicionalmente, estos señores intentan anunciarse gratuitamente aprovechándose de una idea como es Marketclub creado para dar servicio gratis e interesante a todo el club commodoriano nacional... bien... entonces me temo que nos están tomando el pelo a todos, si es que nos lo dejamos tomar.

• Los comerciantes que compran estos productos piratas y se limitan a doblarles el precio, indudablemente venden bastante por debajo del mercado —pero si a beneficios vamos— un 100% no es tampoco nada desdeñable.

• Sí, ya, ¿Pero no decía antes que están dispuestos a anunciarse a precios altos y ahora digo que intentan hacer publicidad gratuita?

También he hablado de mayorías y minorías. Los que quieren anunciarse son algunos de los que les va tan bien el tema que deciden "profesionalizarse" y "establecerse" muchas veces en un apartado de correos y contra reembolso.

6) ¿Qué va a pasar con Mercadillo? Lo hemos dado muchas vueltas. Lo creamos entre todos vosotros y nosotros, con una idea que funciona y que no queremos, en absoluto, destruir por culpa de unos señores "aprovechadillos". Tenemos una lista de una docena de nombres "controlados" y esperemos que no sea necesario ampliarla, pero pedimos la colaboración de todos vosotros.

(Pasa a pág. 77)

COMPARTIENDO EXPERIENCIAS ENTRE AMIGOS



Esta sección está dedicada a la colaboración de todos nuestros lectores y está dividida en dos partes:
1) Programación: Programas y similares.
2) Magia: Trucos, sugerencias, etcétera.
Habrá premios y alicientes para todos los participantes.
Enviarnos vuestra dirección para que podáis poneros en contacto unos con otros.
¡Animaros, chicos (...y chicas)!

C-64

Lectura de números

Jesús M. González Uriarte
C/ Umbe, 5-5º C Drcha.
Guecho (Vizcaya)

Este programa efectúa la transcripción, en letras, del número que se indique, conteniendo nueve caracteres como máximo.

En este programa, fundamentalmente, el número pedido se convierte en una cadena y se determina el número de caracteres que tiene, desglosando cada carácter en unidades, decenas, centenas, unidad de mil, decena de mil, etc., hasta centenas de millón, que es el valor máximo que admite el ordenador en notación standard y se compara, en las sucesivas subrutinas, con las cadenas que indican su equivalencia en letras, imprimiendo, al final, estas cadenas con las variantes de lenguaje correspondientes.

- 1-15 Presentación.
- 17-20 Pone en vacío los caracteres de los números y las cadenas de la equivalencia en letras.
- 22 Se pide el número mediante un INPUT.
- 24 Condiciona el valor máximo permitible.
- 28 Convierte el valor numérico, en una cadena y determina el número de caracteres que tiene.
- 30-60 Desglosa cada carácter del número y lo va comparando con los diferentes bloques de uno, dos, tres y más caracteres, para "leer", respectivamente, unidades, decenas, centenas, etcétera.
- 90-99 Compara y "lee" las unidades.
- 100-123 " " " " decenas.
- 130-150 " " " " centenas.
- 500-502 Variantes del lenguaje.
- 505-530 Imprime los valores menores de mil.
- 600-603 Variantes de lenguaje.
- 605-620 Imprime valores menores de un millón.
- 700-728 Variantes de lenguaje.
- 730-750 Imprime los valores menores de mil millones.
- 1000-1200: Subrutinas.

Indicar, por último, para este programa, que las cantidades de las centenas se han puesto en masculino y cuando se hable de pesetas, por ejemplo, habrá que cambiarlo a femenino. (Por ejemplo, cambiar "cuatrocientos" por "cuatrocientas", solamente para el valor de las centenas y de las centenas de mil).

```
1 REM*****LECTURA DE NUMEROS*******
****
2 REM******** 26/1/1985 ***********
****
3 REM*********** JMGU *************
****
5 PRINT"[CLR]":PRINTTAB(6)"[RVSON]LE
CTURA[SPC]DE[SPC]NUMEROS[RVSOFF]":PR
INT:PRINT:PRINT
6 PRINT"ESTE[SPC]PROGRAMA[SPC]PIDE[SPC]
UN[SPC]NUMERO[SPC]Y[SPC]EFECTUA[SPC]
":PRINT
7 PRINT"LA[SPC]ESCRITURA[SPC]DEL[SPC]
MISMO[SPC]EN[SPC]LETRAS.":PRINT
8 PRINT"SIRVE[SPC]PARA[SPC]IMPRIMIR[SPC]
EN[SPC]LOS[SPC]TALONES[SPC]BAN-":PRI
NT
9 PRINT"CARIOS[SPC]Y[SPC]CARTAS[SPC]
LA[SPC]CANTIDAD[SPC]CORRESPON-[SPC]"
:PRINT
10 PRINT"DIENTE[SPC]EN[SPC]LETRAS[SPC]
Y,TAMBIEN,PARA[SPC]EL":PRINT
11 PRINT"APRENDIZAJE[SPC]DE[SPC]LECT
URA[SPC]DE[SPC]NUMEROS[SPC]EN":PRINT

12 PRINT"CASTELLANO[SPC]PARA[SPC]EXT
RANJEROS.":PRINT
13 PRINT"SE[SPC]HA[SPC]PUESTO[SPC]CO
MO[SPC]LIMITE,MIL[SPC]MILLONES.":PRI
NT
14 GOSUB1000
15 PRINT"[CLR]":FORX=1TO6:PRINT:NEXT
X
17 FORX=0TO3:A$(X)="":B$(X)="":H$(X)
="":NEXTX
20 FORX=0TO3:U$(X)="":D$(X)="":C$(X)
="":NEXTX
22 PRINT"NUMERO[SPC]DESEADO:";:INPUT
N
24 IF N>9.99999999E+8THENGOTO1100
26 C=0:PRINT:PRINT:PRINT:PRINT
28 W$=STR$(N):W=LEN(W$)
30 IFW=2THEN U$(0)=MID$(W$,2,1):GOSU
B90:GOTO500
32 IFW=3THEN U$(0)=MID$(W$,3,1):D$(0
)=MID$(W$,2,1):GOSUB90:GOTO500
34 IFW=4THEN U$(0)=MID$(W$,4,1):D$(0
)=MID$(W$,3,1):C$(0)=MID$(W$,2,1):GO
SUB90
35 IFW=4THENGOTO500
38 IFW=5THEN U$(0)=MID$(W$,5,1):D$(0
)=MID$(W$,4,1):C$(0)=MID$(W$,3,1):GO
SUB90
39 IFW=5THEN U$(1)=MID$(W$,2,1):GOSU
B90:GOTO 600
```

```
42 IFW=6THEN U$(0)=MID$(W$,6,1):D$(0
)=MID$(W$,5,1):C$(0)=MID$(W$,4,1):GO
SUB90
43 IFW=6THEN U$(1)=MID$(W$,3,1):D$(1
)=MID$(W$,2,1):GOSUB90:GOTO600
44 IFW<>7THEN 49
46 IFW=7THEN U$(0)=MID$(W$,7,1):D$(0
)=MID$(W$,6,1):C$(0)=MID$(W$,5,1):GO
SUB90
47 IFW=7THEN U$(1)=MID$(W$,4,1):D$(1
)=MID$(W$,3,1):C$(1)=MID$(W$,2,1):GO
SUB90
48 GOTO 600
49 IFW<>8THEN53
50 IFW=8THEN U$(0)=MID$(W$,8,1):D$(0
)=MID$(W$,7,1):C$(0)=MID$(W$,6,1):GO
SUB90
51 U$(1)=MID$(W$,5,1):D$(1)=MID$(W$,
4,1):C$(1)=MID$(W$,3,1):GOSUB90
52 U$(2)=MID$(W$,2,1):GOSUB90:GOTO 7
00
53 IFW<>9GOTO58
54 IFW=9THEN U$(0)=MID$(W$,9,1):D$(0
)=MID$(W$,8,1):C$(0)=MID$(W$,7,1):GO
SUB90
55 U$(1)=MID$(W$,6,1):D$(1)=MID$(W$,
5,1):C$(1)=MID$(W$,4,1):GOSUB90
56 U$(2)=MID$(W$,3,1):D$(2)=MID$(W$,
2,1):GOSUB90:GOTO 700
58 IFW=10THENU$(0)=MID$(W$,10,1):D$(
0)=MID$(W$,9,1):C$(0)=MID$(W$,8,1):G
OSUB90
59 U$(1)=MID$(W$,7,1):D$(1)=MID$(W$,
6,1):C$(1)=MID$(W$,5,1):GOSUB90
60 U$(2)=MID$(W$,4,1):D$(2)=MID$(W$,
3,1):C$(2)=MID$(W$,2,1):GOSUB90:GOTO
 700
90 IFN=0THENA$(0)="CERO"
91 IFU$(C)="1"THENA$(C)="UN"
92 IFU$(C)="2"THENA$(C)="DOS"
93 IFU$(C)="3"THENA$(C)="TRES"
94 IFU$(C)="4"THENA$(C)="CUATRO"
95 IFU$(C)="5"THENA$(C)="CINCO"
96 IFU$(C)="6"THENA$(C)="SEIS"
97 IFU$(C)="7"THENA$(C)="SIETE"
98 IFU$(C)="8"THENA$(C)="OCHO"
99 IFU$(C)="9"THENA$(C)="NUEVE"
100 IFD$(C)="1"ANDU$(C)="0"THENA$(C)
="DIEZ"
101 IFD$(C)="1"ANDU$(C)="2"THENA$(C)
="DOCE"
102 IFD$(C)="1"ANDU$(C)="3"THENA$(C)
="TRECE"
103 IFD$(C)="1"ANDU$(C)="4"THENA$(C)
="CATORCE"
104 IFD$(C)="1"ANDU$(C)="5"THENA$(C)
="QUINCE"
106 IFD$(C)="1"ANDU$(C)="1"THENA$(C)
="ONCE"
107 IFD$(C)="1"ANDU$(C)>"5"THENB$(C)
="DIECI"
108 IFD$(C)="2"THENB$(C)="VEINTI"
109 IFD$(C)="2"ANDU$(C)="0"THENA$(C)
="VEINTE":B$(C)=""
110 IFD$(C)="3"THENB$(C)="TREINTA[SPC]
Y[SPC]"
111 IFD$(C)="3"ANDU$(C)="0"THENA$(C)
="TREINTA":B$(C)=""
112 IFD$(C)="4"THENB$(C)="CUARENTA[SPC]
Y[SPC]"
113 IFD$(C)="4"ANDU$(C)="0"THENA$(C)
="CUARENTA":B$(C)=""
114 IFD$(C)="5"THENB$(C)="CINCUENTA[SPC]
Y[SPC]"
115 IFD$(C)="5"ANDU$(C)="0"THENA$(C)
="CINCUENTA":B$(C)=""
```

```
116 IFD$(C)="6"THENB$(C)="SESENTA[SPC]
Y[SPC]"
117 IFD$(C)="6"ANDU$(C)="0"THENA$(C)
="SESENTA":B$(C)=""
118 IFD$(C)="7"THENB$(C)="SETENTA[SPC]
Y[SPC]"
119 IFD$(C)="7"ANDU$(C)="0"THENA$(C)
="SETENTA":B$(C)=""
120 IFD$(C)="8"THENB$(C)="OCHENTA[SPC]
Y[SPC]"
121 IFD$(C)="8"ANDU$(C)="0"THENA$(C)
="OCHENTA":B$(C)=""
122 IFD$(C)="9"THENB$(C)="NOVENTA[SPC]
Y[SPC]"
123 IFD$(C)="9"ANDU$(C)="0"THENA$(C)
="NOVENTA":B$(C)=""
130 IFC$(C)="1"THENH$(C)="CIENTO[SPC]
"
131 IFC$(C)="1"ANDD$(C)="0"ANDU$(C)=
"0"THENA$(C)="CIEN":H$(C)=""
132 IFC$(C)="2"THENH$(C)="DOSCIENTOS
[SPC]"
134 IFC$(C)="3"THENH$(C)="TRESCIENTO
S[SPC]"
136 IFC$(C)="4"THENH$(C)="CUATROCIEN
TOS[SPC]"
138 IFC$(C)="5"THENH$(C)="QUINIENTOS
[SPC]"
140 IFC$(C)="6"THENH$(C)="SEISCIENTO
S[SPC]"
142 IFC$(C)="7"THENH$(C)="SETECIENTO
S[SPC]"
144 IFC$(C)="8"THENH$(C)="OCHOCIENTO
S[SPC]"
146 IFC$(C)="9"THENH$(C)="NOVECIENTO
S[SPC]"
149 C=C+1
150 RETURN
500 IFU$(0)="1"THENA$(0)="UNO"
502 IFU$(0)="1"ANDD$(0)="1"THENB$(0)
="[SPC]ONCE[SPC]":H$(0)=""
505 PRINTH$(0);B$(0);A$(0)
510 FORX=1TO4:PRINT:NEXTX
520 GOSUB 1000
530 GOTO 15
600 IF U$(1)="1"ANDD$(1)=""ANDC$(1)=
""THENA$(1)=""
601 IFU$(0)="1"THENA$(0)="UNO"
603 IFU$(0)="1"ANDD$(0)="1"THENB$(0)
="[SPC]ONCE":A$(0)=""
605 PRINTH$(1);B$(1);A$(1);"[SPC]MIL
[SPC]";H$(0);B$(0);A$(0)
610 FORX=1TO6:PRINT:NEXTX
620 GOSUB1000
630 GOTO 15
700 M$="[SPC]MILLONES[SPC]":N$="[SPC]
MIL[SPC]"
710 IF U$(1)="0"ANDD$(1)="0"ANDC$(1)
="0"THEN N$=""
715 IFU$(1)="1"ANDD$(1)="0"ANDC$(1)=
"0"THENA$(1)=""
720 IFU$(2)="1"ANDD$(2)="1"THENB$(2)
="[SPC]ONCE[SPC]":A$(2)=""
725 IFU$(2)="1"ANDD$(2)=""ANDC$(2)="
"THENA$(2)="[SPC]UN[SPC]MILLON[SPC]"
:M$=""
727 IFU$(0)="1"THENA$(0)="UNO"
728 IFU$(0)="1"ANDD$(0)="1"THENB$(0)
="[SPC]ONCE":A$(0)=""
730 PRINTH$(2);B$(2);A$(2);M$;H$(1);
B$(1);A$(1);N$;H$(0);B$(0);A$(0)
740 FORX=1TO5:PRINT:NEXTX
750 GOSUB 1000
760 GOTO 15
1000 PRINTCHR$(18);"PARA[SPC]CONTINU
AR[SPC]PULSAR[SPC]TECLA"
1010 GETX$:IFX$=""THEN1010
1020 RETURN
1100 PRINT"[CLR]":PRINT:PRINT
1120 PRINTTAB(6)"TE[SPC]RECUERDO[SPC]
QUE[SPC]EL[SPC]MAYOR":PRINT
1130 PRINTTAB(6)"NUMERO[SPC]QUE[SPC]
ES[SPC]CAPAZ[SPC]DE[2SPC]":PRINT
1140 PRINTTAB(6)"ESCRIBIR[SPC]DEBE[SPC]
TENER[SPC]NUEVE":PRINT
1150 PRINTTAB(6)"DIGITOS[SPC]COMO[SPC]
MAXIMO.":PRINT
1160 PRINTTAB(6)"ESTE[SPC]ORDENADOR[SPC]
SOLO[SPC]ADMITE":PRINT
1170 PRINTTAB(6)"ESTOS[SPC]DIGITOS[SPC]
EN[SPC]NOTACION[SPC]":PRINT
1180 PRINTTAB(6)"STANDARD.PARA[SPC]V
ALORES[SPC]MAYORES":PRINT
1190 PRINTTAB(6)"SE[SPC]UTILIZA[SPC]
LA[SPC]CIENTIFICA[SPC]QUE":PRINT
1200 PRINTTAB(6)"YA,NO[SPC]TIENE[SPC]
TANTA[SPC]GRACIA."
1210 PRINT:PRINT:PRINT:GOSUB1000:GOT
015
```

VIC-20

Monitor de lenguaje máquina

Víctor Echevarría Ecenarro
C/ Egaña, 5-3º Drcha.
48010 Bilbao

Os envío una nueva colaboración, consistente en un útil programa para el VIC-20, compatible también con el C-64.

Se trata de un pequeño monitor de lenguaje máquina que presenta una importante novedad con respecto a otros programas similares ya publicados: permite introducir los programas **directamente en ensamblador**, evitando el engorro y la pérdida de tiempo y paciencia que supone tener que traducir los mnemónicos y las direcciones hexadecimales a una ristra de números decimales sin sentido aparente.

Se ha intentado dotar al programa de las más necesarias características que ha de poseer un monitor básico, sin intentar ofrecer sofisticaciones no excesivamente útiles que gastarían demasiada memoria. Queden éstas para otros programas comerciales como el cartucho "Machine Languaje Monitor" para el VIC o "Hes Mon 64" para el C-64.

Se han seleccionado, por tanto, cuatro funciones básicas, accesibles a través de un menú que se presenta en pantalla nada más ejecutar el programa:

— Ensamblar (introducir programas en ensamblador)
— Desensamblar (traducir a ensamblador un trozo de memoria)
— Grabar programas en cinta
— Cargar programas desde cinta

Vamos a describir brevemente cada una de estas posibilidades:

—**Ensamblar**: Cuando se pulsa en el menú la opción de ensamblar, el programa pide la dirección hexadecimal inicial y final de entrada del programa. La segunda dirección no hace falta que sea exacta (normalmente uno no sabe exactamente cuántos bytes va a gastar). Una vez dados los datos, el programa escribe la primera dirección y aguarda a que el usuario teclee la correspondiente instrucción en ensamblador; cuando se pulsa "return" el programa valora la instrucción, y si no es correcta escribe "error", y permite una nueva entrada. La instrucción puede ir con o sin espacio de separación entre el mnemónico y el resto de la línea. Ejemplo: el programa entiende igual LDA$EA27 que LDA $EA27.

Una vez aceptada una línea el programa escribe la dirección siguiente y espera al usuario. Se continúa así hasta que por fin, cuando el programa se ha terminado de introducir, se teclea la palabra "FIN". Entonces el programa dará por terminado el proceso de ensamblaje, y presentará una nueva demanda de entrada de otra dirección inicial, por si se deseara continuar ensamblando en otras zonas de memoria. Si se pulsa "return" se vuelve al menú.

NOTA: Aunque el programa detecta errores cometidos en la entrada no conviene hacer entradas absurdas como por ejemplo introducir una cadena de menos de tres caracteres o algo similar

"para ver qué pasa", porque en este caso el programa se interrumpirá por un illegal quantity, y habrá que activarlo de nuevo, lo cual no es muy agradable.

—**Desensamblar:** Cuando se elige esta opción el programa pide dirección inicial y final (en hexadecimal) y procede seguidamente al desensamblado. Si se pulsa una tecla, el programa interrumpe el listado hasta que se pulse otra cualquiera (esto resulta útil si se desea copiar algunas instrucciones antes de que desaparezcan). Si la segunda tecla pulsada es "return", el programa pide de nuevo una dirección inicial, por si se desea desensamblar otras áreas de memoria. En caso negativo un nuevo "return" nos devolverá al menú.

—***Grabar en cinta:*** Al elegir grabar un programa que tengamos en memoria se piden las direcciones inicial y final del programa y el nombre que queremos ponerle, y se graba *sin más problemas.*

—**Cargar de cinta:** Se pide el nombre del programa que queremos cargar, y cuando se localiza en la cinta se carga **en las mismas direcciones en que fue grabado.**

Estas dos últimas opciones podían haberse realizado mediante las rutinas SETLFS, SETNAM, LOAD, y SAVE del Kernal, pero desde Basic es más sencillo e igual de efectivo.

El programa fue escrito para un VIC-20 *con* 3K de expansión, aunque vale igual con el superexpander, o con cualquier otro cartucho (8K o 16K). De hecho, cuanta más memoria se tenga, de tanto más espacio se dispondrá para situar los programas que se creen en lenguaje máquina, sin invadir el Basic. (Si se usa sólo el cartucho de 3K conviene proteger el área que deseemos utilizar para los programas, con objeto de evitar sorpresas desagradables).

Pienso que el programa vale también para un C-64, cambiando un número de la línea 135: en la parte final de esta línea dice: "IF PEEK(197)=15 THEN...". Este quince significa la tecla "return" en el VIC-20. En el C-64 hay que cambiarlo por un uno. No he comprobado el funcionamiento del programa en un C-64, pero como no se hace referencia a direcciones específicas del VIC no debería haber problemas. En todo caso, cualquier usuario del C-64 familiarizado *mínimamente con su máquina* puede solventar cualquier pequeña dificultad que pudiera persistir, y que yo he pasado por alto.

Por si alguien quiere añadir alguna opción más (como volcado hexadecimal, cambio de base de números, o ejecución del código máquina) que yo no he considerado suficientemente interesantes por la relación utilidad que presentan-memoria que gastan (dado que conviene que haya el máximo de memoria libre para los programas en lenguaje máquina), aquí señalo algunas partes importantes del programa:

9—Declara la matriz P$, y carga en ella los gráficos de los trece modos de direccionamiento.

65—Línea de entrada de las instrucciones en ensamblador. No se ha utilizado un *simple* "input" porque al introducir una instruccin en algún modo de direccionamiento indexado sólo aceptaría la cadena anterior a la coma, dando un extra ignored por el resto. Ej.: *si se mete* STA $22FC,X aceptaría STA $22FC e ignoraría el resto.

75—Si ha habido algún error la variable FL estará a uno.

90-125—Interpretan la instrucción y deciden su modo de direccionamiento.

250, 260, 270, 280, subrutinas auxiliares de la 90.

130-255—Desensamblado.

300—Cambio hex-dec.

350—Cambio dec-hex.

400—Crea fichero en la cinta.

470—Carga fichero (detecta errores por la variable ST).

500-540—Mnemónicos del 6502/6510 seguidos de una letra que designa su modo de direccionamiento.

600-610—Cadenas (que se cargan en la matriz P$) que son los gráficos de los distintos modos de direccionamiento.

Considero que este monitor de lenguaje máquina constituye una útil herramienta para quienes nos iniciamos en este lenguaje, y es una buena ayuda para introducirnos en los entresijos de nuestras máquinas y conocerlas un poco más a fondo.

```
0 REM (C) VICTOR ETXEBARRIA 1985
9 DIMP$(12):FORI=0TO255:READE$:NEXT:
FORI=0TO12:READP$(I):NEXT:RESTORE
10 PRINT"[CLR][RED][RVSON][4CRSRR][2SPC]
MONITOR[SPC]DE[2SPC][CRSRD][RVSOFF]"
:PRINT"[RED][RVSON][CRSRR][2SPC]LENG
UAJE[SPC]MAQUINA[2SPC][CRSRD][BLU][RVSOFF]
"
11 PRINT"[7CRSRR]-MENU-[2CRSRD]"
```

```
15 PRINT"[CRSRR]1)ENSAMBLADO":PRINT"
[CRSRR]2)DESENSAMBLADO"
20 PRINT"[CRSRR]3)GRABAR[SPC]EN[SPC]
CINTA":PRINT"[CRSRR]4)CARGAR[SPC]DE[SPC]
CINTA":PRINT"[CRSRR]5)FIN[CRSRD]"
25 GETA$:IFA$=""THEN25
27 TT=VAL(A$):IFTT<1ORTT>5THEN25
30 IFTT=5THENEND
35 IFTT=4THENINPUT"NOMBRE";W$:GOTO47
0
40 B=0:I$="":INPUT"DESDE";I$:IFI$=""
THEN10
45 L$=I$:GOSUB300:I=L
50 INPUT"HASTA";S$:L$=S$:GOSUB300:S=
L
55 ONTTGOTO60,130,400
60 FORK=ITOS:I=I+B:IFI>STHEN80
62 K=I:L=I:GOSUB350:I$=L$
63 PRINTI$+"[SPC]";:R$=""
65 GETX$:IFX$<>CHR$(13)THENPRINTX$;:
IFX$<>CHR$(20)THENR$=R$+X$:GOTO65
66 IFX$=CHR$(20)THENR$=LEFT$(R$,LEN(
R$)-1):GOTO65
67 PRINTCHR$(13);
69 IFR$="FIN"THENK=S:GOTO80
70 GOSUB90
75 IFFLTHENFL=0:PRINT"ERROR.":GOTO63

80 NEXT
85 GOTO40
90 T=LEN(R$):RR$=LEFT$(R$,3)
91 ONT-2GOTO93,95,95,97,97,100,100,1
10,110
92 FL=1:RETURN
93 Q$="A":B=1:GOSUB250:GOSUB260:RETU
RN
95 IFRIGHT$(R$,1)="A"THENQ$="B":B=1:
GOSUB250:GOSUB260:RETURN
96 GOTO92
97 IFMID$(R$,T-3,1)="#"THENQ$="D":B=
2:GOSUB750:RETURN
98 IFMID$(R$,T-2,1)="$"THENQ$="C":B=
2:GOSUB750:RETURN
99 GOTO92
100 IFMID$(R$,T-3,1)="#"THENQ$="D":B
=2:GOSUB750:RETURN
101 IFLEFT$(R$,1)="B"ANDRR$<>"BIT"TH
ENQ$="I":B=2:GOSUB700:RETURN
103 IFRIGHT$(R$,2)=",X"THENQ$="E":B=
2:T=T-2:GOSUB750:RETURN
105 IFRIGHT$(R$,2)=",Y"THENQ$="F":B=
2:T=T-2:GOSUB750:RETURN
107 IFMID$(R$,T-4,1)="$"THENQ$="J":B
=3:GOSUB750:RETURN
108 GOTO92
110 IFRIGHT$(R$,2)=",X"THENQ$="K":B=
3:T=T-2:GOSUB750:RETURN
113 IFRIGHT$(R$,3)="),Y"THENQ$="H":B
=2:T=T-3:GOSUB750:RETURN
115 IFRIGHT$(R$,2)=",Y"THENQ$="L":B=
3:T=T-2:GOSUB750:RETURN
117 IFRIGHT$(R$,3)=",X)"THENQ$="G":B
=2:T=T-3:GOSUB750:RETURN
123 IFMID$(R$,T-6,1)="("THENQ$="M":B
=3:T=T-1:GOSUB750:RETURN
125 GOTO92
130 FORK=ITOS:I=I+B:IFI>STHEN140
133 K=I:L=K:GOSUB350:I$=L$:GOSUB150
135 PRINTI$+"[SPC]"+R$:IFPEEK(198)>0
THENPOKE198,0:WAIT198,1:POKE198,0:IF
PEEK(197)=15THENK=S
140 NEXT:GOTO40
150 T=PEEK(I):FORX=0TOT:READE$:NEXT:
RESTORE:IFE$=""THENE$="???A":B=1
155 RR$=LEFT$(E$,3):N$=RIGHT$(E$,1):
Z=ASC(N$)-65:IFZ<2THENR$=RR$+P$(Z):B
=1:RETURN
160 IFZ>8THENB=3:GOTO175
163 B=2
175 FORD=B-1TO1STEP-1:Y=PEEK(I+D):V=
2:IFLEFT$(E$,1)="B"ANDRR$<>"BIT"THEN
GOSUB800
178 L=Y:GOSUB350:Y$=RIGHT$(L$,V):U$=
U$+Y$:NEXT
180 R$=RR$+P$(Z)+U$:U$="":RETURN
250 FORX=0TO255:READE$:IFE$=RR$+Q$TH
ENPOKEI,X:BA=1:X=256
255 NEXT:RESTORE:RETURN
260 IFBATHENBA=0:RETURN
265 FL=1:RETURN
270 FORO=1TOB-1:N$=MID$(R$,(T+1)-2*O
,2):L$=N$:GOSUB300:POKEI+O,L
275 NEXT:RETURN
280 N$=RIGHT$(R$,4):L$=N$:GOSUB300:B
R=L
285 IFBR>I+1ANDBR-1-I<128THENPOKEI+1
,BR-I-2:GOTO295
290 IFBR=<I+1ANDI+1-BR<129THENPOKEI+
1,255-I-1+BR:GOTO295
293 FL=1
295 RETURN
300 H=LEN(L$):L=0:FORC=H-1TO0STEP-1
310 W=ASC(MID$(L$,H-C,1))-48:IFW>9TH
ENW=W-7
320 L=L+W*16↑C:NEXT
330 RETURN
350 L$="":FORH=0TO3:G=INT(L/16):A=L-
16*G
360 L$=CHR$(A+48-7*(A>9))+L$:L=G:NEX
T:RETURN
400 INPUT"QUE[SPC]NOMBRE";W$
410 OPEN9,1,1,W$
415 PRINT"GRABANDO[SPC]";W$
420 PRINT#9,I
430 FORK=ITOS
440 PRINT#9,PEEK(K)
450 NEXT
455 CLOSE9:PRINT"PROGRAMA[SPC]GRABAD
O.":FORCH=0TO2000:NEXT:GOTO10
470 OPEN9,1,0,W$
471 PRINT"CARGANDO[SPC]";W$
473 INPUT#9,I:J=0
475 INPUT#9,K
477 IFST<>0THEN485
480 POKEI+J,K:J=J+1:GOTO475
485 IFST=64THENPOKEI+J,K:GOTO495
490 PRINT"ERROR[SPC]EN[SPC]LA[SPC]CI
NTA.":FORCH=0TO2000:NEXT:GOTO10
495 CLOSE9:PRINT"PROGRAMA[SPC]CARGAD
O.":FORCH=0TO2000:NEXT:GOTO10
500 DATABRKA,ORAG,,,,ORAC,ASLC,
503 DATAPHPA,ORAD,ASLB,,,ORAJ,ASLJ
505 DATA,BPLI,ORAH,,,,ORAE,ASLE,,CLC
A,ORAL,,,,ORAK
507 DATAASLK,,JSRJ,ANDG,,,BITC,ANDC,
ROLC,,PLPA,ANDD
510 DATAROLB,,BITJ,ANDJ,ROLJ,,BMII,A
NDH,,,,ANDE,ROLE,
513 DATASECA,ANDL,,,,ANDK,ROLK,,RTIA
,EORG,,,,EORC
515 DATALSRC,,PHAA,EORD,LSRB,,JMPJ,E
ORJ,LSRJ,,BVCI,EORH,,,,EORE
517 DATALSRE,,CLIA,EORL,,,,EORK,LSRK
,,RTSA,ADCG,,,,ADCC,RORC,,PLAA
520 DATAADCD,RORB,,JMPM,ADCJ,RORJ,,B
VSI,ADCH,,,,ADCE,RORE,,SEIA
523 DATAADCL,,,,ADCK,RORK,,,STAG,,,S
TYC,STAC,STXC,,DEYA,,TXAA,,STYJ
525 DATASTAJ,STXJ,,BCCI,STAH,,,STYE,
STAE,STXMONITOR L.M.,TXSA,,,STAK
527 DATA,,LDYD,LDAG,LDXD,,LDYC,LDAC,
LDXC,,TAYA,LDAD,TAXA,,LDYJ,LDAJ
530 DATALDXJ,,BCSI,LDAH,,,LDYE,LDAE,
LDXF,,CLVA,LDAL,TSXA,,LDYK,LDAK
533 DATALDXL,,CPYD,CMPG,,,CPYC,CMPC,
```

```
DECC,,INYA,CMPD,DEXA,,CPYJ,CMPJ
535 DATADECJ,,BNEI,CMPH,,,,CMPE,DECE
,,CLDA,CMPL,,,,CMPK,DECK,,CPXD
537 DATASBCG,,,CPXC,SBCC,INCC,,INXA,
SBCD,NOPA,,CPXJ,SBCJ,INCJ,,BEQI
540 DATASBCH,,,,SBCE,INCE,,SEDA,SBCL
,,,,SBCK,INCK,
600 DATA,"[SPC]A","[SPC]$","[SPC]#$"
,"[SPC]$[2SPC],X[4CRSRL]","[SPC]$[2SPC]
,Y[4CRSRL]","[SPC]($[2SPC],X)[5CRSRL]
","[SPC]($[2SPC]),Y[5CRSRL]"
610 DATA"[SPC]$","[SPC]$","[SPC]$[4SPC]
,X[6CRSRL]","[SPC]$[4SPC],Y[6CRSRL]"
,"[SPC]($[4SPC])[5CRSRL]"
700 GOSUB250:GOSUB280:GOSUB260:RETUR
N
750 GOSUB250:GOSUB270:GOSUB260:RETUR
N
800 V=4:IFY<128THENY=I+1+D+Y:RETURN
810 Y=I+D-255+Y:RETURN
```

C-64

Busca + List y grabar C/M

Guillermo Som Cerezo
C/ Animas, 2-1º Izqda.
Nerja (Málaga)

Soy fotógrafo y tengo una tienda (pequeña) para vender carretes y otras cosillas.

Tengo un Commodore 64 desde mayo de 1983, antes tuve un VIC-20 (desde enero 1983).

Lo uso para todo: existencias, contabilidad, juegos y sobre todo para acostarme después de las tres, algunas noches.

La revista, en general, está bien, pero noto que falláis mucho en los listados, hay **"demasiados"** errores de imprenta. Y algunas veces cuando lo rectificáis, no lo hacéis bien, al menos eso me parece en el programa de conversión del Kernal, aparecido en el nº 5 y rectificado en el nº 6.

Creo que en la línea 220 no debería estar el **D=D+1**, al menos, cuando lo quité me funcionó el programa.

Pero en fin, fallos tenemos todos y siempre se procura hacerlo bien.

Otra cosilla que no veo bien es que publiquéis un programa que con anterioridad ha aparecido en otra revista.

En el caso del procesador de textos del último número, (diciembre nº 10).

Este mismo programa, en versión inglesa, apareció en Micro-Sistemas, precisamente en el número de febrero, con el cual salió la revista vuestra.

Busca + List

Os envío un programa compuesto de dos rutinas que pueden ser de utilidad a la hora de corregir y repasar los listados de esos programas largos, que casi siempre nos dan problemas, bien por haber omitido algo, bien por haber puesto algo de más.

La primera parte del programa que os mando es un "busca líneas". La función que hace, es precisamente eso, buscar líneas tras las instrucciones goto, gosub y then.

Esta primera parte incluye las líneas 63000 a 63150.

La segunda rutina es para listar los programas. Podréis decir que es más fácil usar el **"list"**, pero las ventajas que se obtienen con esta rutina son las siguientes:

Podemos ir viendo (o listando) línea por línea hasta que pulsemos una tecla, nos evitamos así tener de detener el listado normal o el ir listando cada línea por vez.

Además, tenemos la ventaja de que los caracteres se vayan imprimiendo poco a poco gracias al bucle de retardo de la línea 63750 la rapidez de imprimir los caracteres depende del valor contenido en la variable **PA**. Cuanto más alto sea el número, más lentamente se irán imprimiendo los caracteres.

Podemos controlar el valor del "PA" bien modificando el que se le asigna en la línea 63500, o bien al pulsar la tecla de función f1 la reducimos en 10 unidades, o bien al pulsar f7 lo aumentamos en 10.

Líneas 63782 y 63784.

Pero no acaba aquí la cosa.

También podemos ir listando **"hacia atrás"** pulsando el signo (—) menos.

El programa listar va desde la línea 63500 a la 63950, aquí se incluye una subrutina para asignar las palabras Basic a la variable TK$(x), para luego convertir los tokens en palabras.

Esta subrutina está entre las líneas 63900 a 63940.

Este programa se puede añadir a otro ya existente en la memoria cambiando los punteros de inicio del Basic:

Poke 43, Peek (45)-2: Poke 44, Peek (46) se carga estas rutinas y se vuelve a dar los valores originales a estas dos posiciones, que suelen ser 1 para la posición 43 y 8 para la 44.

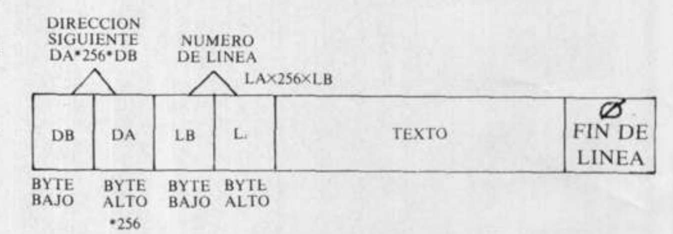
Para poder usar estas rutinas en otro ordenador distinto al C-64, se han de modificar los valores de SA.

En la línea 63010 haciendo SA=Peek(44)*256+Peek(43)-1

Y en la 63500 con SA=Peek(44)*256+Peek(43) y en la línea 63770 cambiar el valor 1904 por el correspondiente a la dirección de la pantalla (8142 para el VIC 20 no ampliado).

El principio de estas dos rutinas está en la forma en que los Commodore almacenan las líneas de los programas.

Es la siguiente:



Los dos primeros bytes señalan la siguiente dirección de memoria en que se encuentra la línea siguiente, seguidos de otros dos bytes con el número de línea, a continuación vienen las instrucciones y el texto para terminar con un (Ø) cero, como indicador de fin de línea.

Si la primera línea de programa fuese:

1Ø PRINT "HOLA"

Se almacenaría:

13	8	10	0	153	34	72	79	76	65	34	Ø
SIGUIENTE DIRECCION		NUM. DE LINEA		PRINT	"	H	O	L	A	"	FIN DE LINEA
2061		1Ø									

Si el programa terminara con esta línea, las posiciones 2Ø61 y 2Ø62 contendrían un (Ø) cero cada una, indicando así el final del programa.

Todo esto ya ha sido antes explicado en "Ventana CBM" del nº 5 del Club Commodore, pero creo que no viene mal recordarlo, así como puede ser de utilidad para los que se han unido recientemente al "mundillo" de los Commodore.

Perdonad que me "enrolle" tanto, pero es la primera vez que os escribo, y me estoy aprovechando un poco.

Pero es que lo que voy a decir viene al caso.

Y es un "truco" para modificar los números de líneas y confundir a quien liste el programa. Supongamos que tenemos este programa:

```
1Ø FOR X=1 TO 1Ø
2Ø PRINT "HOLA"
3Ø NEXT X
```

La direcciones 2Ø51-2Ø52, 2Ø63-2Ø64 y 2Ø75-2Ø76 contienen los valores de los números de línea en 2Ø51=1Ø, 2Ø63=2Ø y 2Ø75=3Ø.

Si hacemos Poke 2Ø63,1Ø:Poke 2Ø75,1Ø y listamos el programa, para nuestra ¿sorpresa? las tres líneas anteriores, ahora tienen el mismo número.

¿Qué ocurre si lo ejecutamos?, probemos: vemos que funciona correctamente.

¿Qué os parece un programa con números de líneas iguales? (un desastre sería).

También este programa sirve para grabar los programas de código máquina almacenados en el ordenador. Para volver a instalarlos en el ordenador, usar Load "nomb.", periférico, 1.

Para usarlo hay que dar la dirección de inicio y la dirección final, más uno.

Por ejemplo podemos grabar el DOS WEDGE, dando:

Dirección inicio=52224

Dirección final+1=53081

Nombre="DOS 5.1".

BUSCA + LIST

```
63000 REM BUSCA DIRECCIONES TRAS GOT
O, GOSUB, THEN
63010 SA=2048:S1=SA
63020 INPUT"[CRSRD]NUMERO[SPC]DE[SPC]
LINEA[SPC]A[SPC]BUSCAR:";VA
63030 S2=PEEK(S1+2)*256+PEEK(S1+1)-1
:S3=S1+4:S4=S2+1
63040 NL=PEEK(S1+4)*256+PEEK(S1+3)
63050 FORX=S3TOS4:D=PEEK(X)
63059 REM*** SI NO ES NI GOTO,GOSUB
O THEN ENTONCES A LA SIGUIENTE POSIC
ION
63060 IFD<>137ANDD<>141ANDD<>167THEN
63120
63070 FORY=XTOS4:D1=PEEK(Y):IFD1=58T
HEN63090:REM*** BUSCA DOS PUNTOS
63080 NEXTY
63090 A$="":FORZ=X+1TOY-1:A$=A$+CHR$
(PEEK(Z)):NEXTZ
63100 NU=VAL(A$):IFNU<>VATHEN63120
63110 PRINT"[CRSRD]"NU;"ESTA[SPC]EN[SPC]
LA[SPC]LINEA:";PEEK(S1+4)*256+PEEK(S
1+3)
63120 NEXT
63130 S1=S2:IFS1<1THENEND
```

NOTICIAS COMMODORE

Curso B.C.P. en el Club de Programación Alaiz

El pasado día 5 de marzo comenzó en el Club de Programación Alaiz de Pamplona el curso denominado "Beginner's Computing Program" dirigido a los profesionales que busquen una útil introducción en el mundo de la informática y ofimática.

Las sesiones se imparten los martes y jueves a última hora de la tarde. Los primeros días serán una introducción al lenguaje B.A.S.I.C. Posteriormente se pasa al conocimiento de un paquete de software integrado, en concreto el Lotus 1,2,3. Los jueves, por su parte, se trata de plantear diversos temas sobre aplicaciones concretas de los ordenadores personales.

Los profesores son especialistas en sus diversos campos.

(Nota de Redacción: Agradecemos el interés que ha demostrado siempre el Club Alaiz de Pamplona en enviarnos sus noticias y programas. No os preocupéis, ya irán apareciendo).

Club Alaiz
Plaza Monasterio Santa Gema, s/n
Telf.: 254480/257704
Pamplona

Programa de informática en Radio Popular de Vigo

Desde el mes de febrero se está emitiendo por la onda media de Radio Popular de Vigo un programa titulado "Novas de Informática", dedicado a la informática y organizado por el Club de Microprocesadores del Círculo Ourensan-Vigues. El programa pretende informar sobre el mundo de la informática, fundamentalmente de los ordenadores personales. Está redactado en gallego, y tiene una emisión semanal, todos los martes a las 9 de la noche. Todos los componentes del equipo de redacción no son profesionales de la radio y están unidos al Club de Microordenadores desde su creación.

```
63140 REMPRINT"[CRSRD]LINEA:";NL
63150 GOTO63030
63200 :
63300 :
63400 REM ***************************
**
63410 REM ***      LISTADOS          *
**
63420 REM ***                        *
**
63430 REM *** GUILLERMO SOM CEREZO *
**
63440 REM ***                        *
**
63450 REM *** NERJA  9/10 DIC 1984 *
**
63460 REM ***************************
**
63470 :
63480 :
63499 REM*** INICIAR
63500 SA=2049:S1=SA:GOSUB63600:GOSUB
63900:PA=30:REM PAUSA
63520 GOTO63700
63599 REM*** CABECERA
63600 PRINT"[CLR]"SPC(7);"[CRSRD][RVSON]
[3SPC]*****[SPC]LISTADOS[SPC]*****[3SPC]
[RVSOFF][CRSRD]":RETURN
63700 REM *** PARTE PRINCIPAL ***
63710 S2=PEEK(S1)+PEEK(S1+1)*256:NL=
PEEK(S1+2)+PEEK(S1+3)*256
63720 PRINT"[CRSRL]";NL;:FORX=S1+4TO
S2-1:D=PEEK(X):IFD=0THENPRINT:GOTO63
750
63722 IFD=34THEND1=D1+D:IFD1>DTHEND1
=0
63723 REM*** SI HAY COMILLAS NO ES U
N TOKEN
63724 IFD1=34THENIFD>127THEN63740
63729 REM*** SI ES TOKEN IMPRIMIR TK
$
63730 IFD>127ANDD<204THENPRINTTK$(D-
128);:GOTO63750
63740 PRINTCHR$(D);
63749 REM*** BUCLE PARA LA PAUSA
63750 FORI=0TOPA:NEXTI:NEXT:D1=0
63760 GETA$:IFA$=""THEN63760
63769 REM*** SI SE LLEGA A LA ANTEPE
NULTIMA LINEA DE PANTALLA, BORRARLA.

63770 IFPEEK(1984)<>32THENGOSUB63600

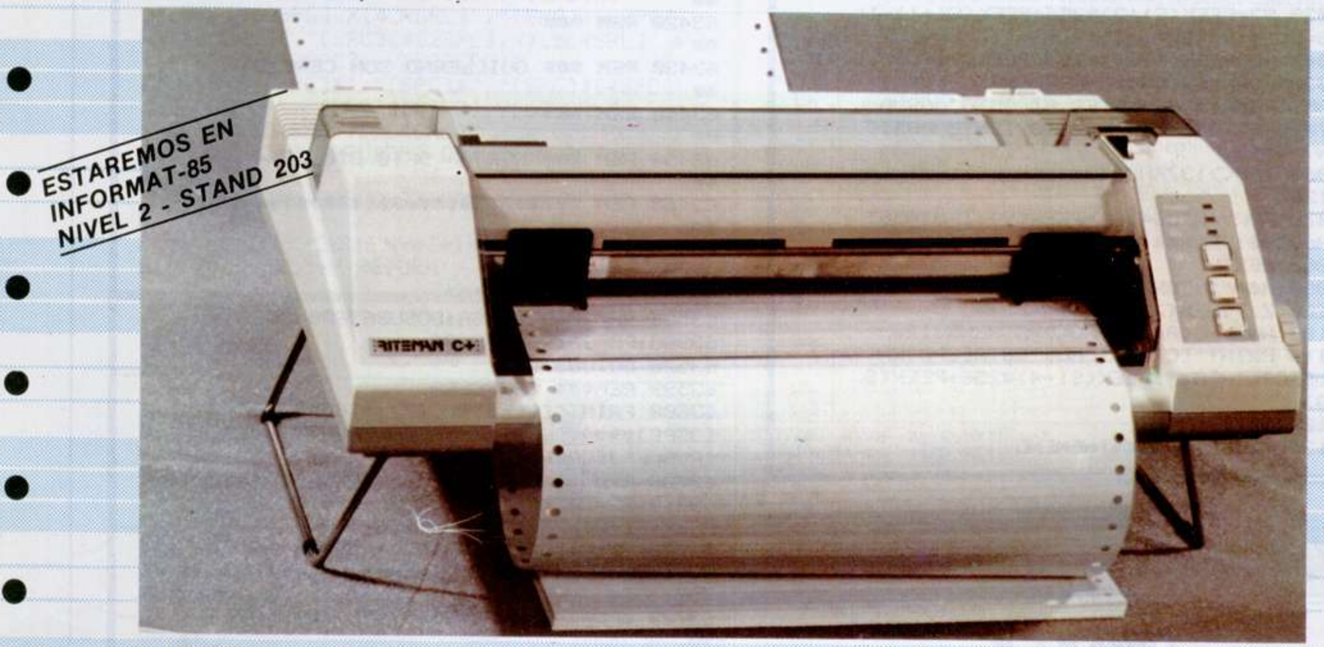
63780 IFA$="-"THEN63820:REM RETROCED
E
63781 REM*** F1
63782 IFA$="[F1]"THENPA=PA-10:IFPA<0
THENPA=0
63783 REM*** F7
63784 IFA$="[F7]"THENPA=PA+10:IFPA>1
00THENPA=100
63790 S1=S2:REM AVANZA
63799 REM*** SI SE LLEGA A LA ULTIMA
 LINEA, TERMINAR
63800 IFS1=PEEK(46)*256+PEEK(45)-2TH
ENEND
63810 GOTO63710
63819 REM*** SI SE PULSA '-' RETROCE
DE
63820 IFS1=SATHEN63760
63830 S1=S1-2:IFPEEK(S1)=0ANDPEEK(S1
-1)<>0ANDPEEK(S1-2)<>0THENS1=S1+1:GO
TO63800
63840 S1=S1+1:GOTO63820
63899 REM*** ALMACENA LAS PALABRAS B
ASIC EN LA VARIABLE TK$ PARA LOS TOK
ENS
```

:RITEMAN:

PERSONAL/BUSINESS PRINTER

AMPLIA GAMA

Nuevas impresoras modelos **F+** y **C+**, sin rodillo alimentación horizontal, impresión vertical, tracción y fricción desde 4 a 10", bidireccional optimizada velocidad 105 cps. con soportes de elevación.

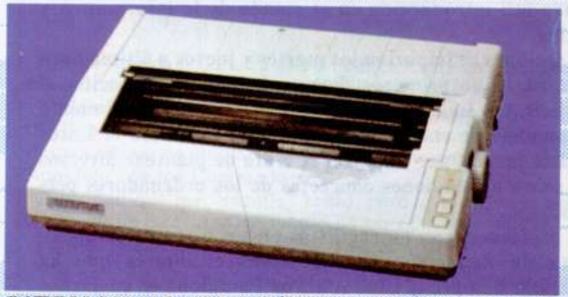


RITEMAN F+: Interface Paralelo Centronics, 2K buffer NLQ — **P.V.P. 69.000 pts.**
RITEMAN C+: Especial directa a COMMODORE (cable inc.) — **P.V.P. 67.000 pts.**

Otros modelos RITEMAN en 80 y 136 columnas, velocidad 120, 140, 160 cps.



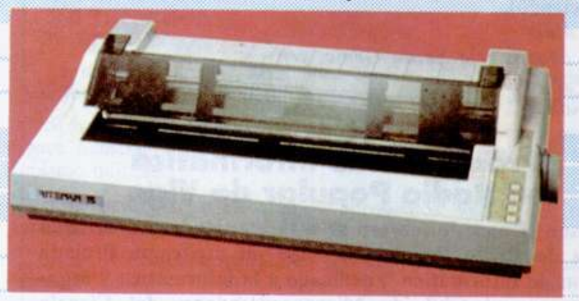
RITEMAN 10, 120 cps. P.V.P. 81.000



RITEMAN 10-IBM, 140 cps. P.V.P. 85.000



RITEMAN 10-II 160 cps. P.V.P. 93.000



RITEMAN 15 160 cps. P.V.P. 155.000

DE VENTA EN LOS MEJORES ESTABLECIMIENTOS ESPECIALIZADOS



DATAMON, S. A.

PROVENZA, 385-387, 6.º, 1.ª
TELÉFONO (93) 207 27 04*

REPRESENTACION EN ESPAÑA DE:

:RITEMAN:

-IMPRESORAS PROFESIONALES-

08025 - BARCELONA

* **MAYORES PRESTACIONES**
* **MENOR TAMAÑO**
* **MEJOR PRECIO**

```
63900 I=0:DIMTK$(75)
63910 FORN=41118TO41373:K=PEEK(N)
63920 IFK<128THENA$=A$+CHR$(K)
63930 IFK>=128THENA$=A$+CHR$(K-128):
TK$(I)=A$:I=I+1:A$="":REMPRINTI-1;TK
$(I-1)
63940 NEXT:RETURN
63950 END
```

GRABAR C/M

```
0 REM *********************
1 REM *** GUILLERMO SOM ***
2 REM ***  NERJA  1984  ***
3 REM *********************
4 :
5 :
10 PRINT"[CLR][CRSRD]***[SPC]GRABAR[SPC]
PROGRAMAS[SPC]EN[SPC]C/M[SPC]***"
20 INPUT"[CRSRD]NOMBRE[SPC]DEL[SPC]P
ROGRAMA";A$:L=LEN(A$):IFL>16THEN10
30 INPUT"[CRSRD]DIRECCION[SPC]INICIO
";DI:IFDI<0ORDI>65535THEN30
40 INPUT"[CRSRD]SIGUIENTE[SPC]DIRECC
ION[SPC]DISPONIBLE";D:IFD<=DITHEN40
50 PRINT"[2CRSRD]PREPARA[SPC]LA[SPC]
CINTA[SPC]O[SPC]EL[SPC]DISCO":POKE19
8,0:WAIT198,1
52 IFPEEK(186)=1THEN60:REM SI ES CAS
ET
53 REM *** INICIALIZA EL DISCO Y LEE
 EL CANAL DE ERRORES ***
54 OPEN15,8,15,"I":INPUT#15,EN,E$,ET
,ES:CLOSE15
56 PRINTEN;E$;ET;ES:IFEN<>0THENSTOP
60 FORN=1TOL:POKE678+N,ASC(MID$(A$,N
,1)):NEXT
70 POKE780,L:POKE781,167:POKE782,2
72 REM*** RUTINA KERNAL SETNAM ***
80 SYS65469
90 POKE254,INT(DI/256):POKE253,DI-PE
EK(254)*256:POKE780,253
100 POKE782,INT(D/256):POKE781,D-PEE
K(782)*256
102 REM*** RUTINA KERNAL SAVE ***
110 SYS65496
120 END
```

VIC-20 con ampliación

Banco de datos

Miquel Gras Carreras
Pl. Cots, 3. 6º-3º.
Manresa (Barcelona)

Me llamo Miquel, tengo 16 años y vivo en Manresa. Os envío una nueva colaboración a esta maravillosa revista que es Commodore World, esta vez no es un juego, sino un interesante banco de datos para el VIC-20.

Este programa ocupa unos 2920 bytes, con lo cual se puede utilizar en un VIC sin ampliación, pero una vez el programa contiene los datos, esta cifra aumenta, con lo cual si no se tiene ampliación, se podrán almacenar pocos datos en un fichero, pero haciendo varios ficheros, el problema se soluciona.

Ahora paso a dar las instrucciones de su uso:

El programa al iniciarse presenta en la pantalla varias opciones (menú), del cual elegiremos.

Si introducimos datos al ordenador nos pregunta cuántos nombres (personas) queremos en el fichero, la respuesta dependerá de la memoria que tenga nuestro VIC, para un VIC sin ampliación podría ser 5 ó 6, como máximo 7, ya que el ordenador de cada persona almacenará:

—Nombre, dirección, telefóno, relación (de qué le conocemos), otros.

Al introducir los datos, el ordenador nos irá preguntando de cada persona lo mencionado, y al terminar volveremos a tener en pantalla el menú.

En las opciones 4 (búsqueda de dato) y 8 (cambio de dato) se ha de entrar el número que ocupa la persona en el fichero, no su nombre. Este número lo obtendremos pulsando 3, que nos clasificará los nombres de las personas con el número que ocupan en el fichero.

Si se ha de alterar un dato de una persona, se pulsa 8 y el número que ocupa la persona en el fichero, seguidamente, el ordenador nos preguntará los nuevos datos, para almacenarlos en el cassette habrá que rebobinar la cinta hasta el principio del fichero y pulsar 5 (salvar datos), y estos nuevos datos quedarán grabados sobre los viejos.

Y para volver a introducir los datos del cassette al ordenador pulsar 6 (cargar datos), el nombre del fichero y cuantos datos hay en ese fichero.

Ahora paso a daros algunos consejos para el programa.

—Utilizar una cinta de 20 minutos exclusivamente para el banco de datos.

—Al principio de la cinta hacerse una cabecera, en la cual se pueda almacenar el nombre del fichero, número de datos del fichero y en la vuelta de contador en que se encuentra.

—Dejar unas 10 vueltas entre fichero y fichero.

—Si no queréis que el programa vaya a esta velocidad, podéis suprimir los pokes de las líneas: -110, 740, 940, 1080 y 1210.

Perdonad tanto rollo, pero creo que es un programa del cual se han de dar unas instrucciones para su funcionamiento correcto. Y si os quedara alguna duda sobre el programa, escribidme y procuraré aclarárosla.

GLOSARIO

Device - Dispositivo. Un componente eléctrico, electrónico o mecánico de un sistema.

Diagnostic routine - Rutina de diagnóstico. Es una rutina destinada a la detección de errores.

Dialog - Diálogo. Una secuencia de preguntas y respuestas en un sistema interactivo, es similar a una conversación entre personas.

Dichotomizing search - Búsqueda dicotómica. Es un sistema de búsqueda entre un grupo ordenado de elementos basado en dividir en dos el grupo, rechazando la parte que no contiene el elemento buscado, este proceso se repite con la parte aceptada hasta que la búsqueda termina. Se denomina también búsqueda binaria.

Dichotomy - Dicotomía. División en subórdenes o subclases, por ejemplo: blancos o no blancos, ceros o no ceros, etc.

Digit - Dígito. Un símbolo que representa un entero no negativo y menor que la base tomada, por ejemplo en base 2 el 0 ó el 1, o en base 10 cualquiera de los números del 0 al 9.

Digital. Relativo al manejo de datos en forma de dígitos.

Digital/analog converter - Conversor digital/analógico. Es un dispositivo capaz de convertir una señal digital en otra de tipo analógica.

DIR. Directorio.

BANCO DE DATOS

```
10 REM MIQUEL GRAS
20 REM
30 REM CARRERAS (C)
40 REM
50 REM MANRESA 1984
60 REM
70 PRINT"[CLR]"
80 POKE36879,232
90 GOSUB740
100 END
110 POKE37158,38:POKE37159,72
120 PRINT"[CLR][2CRSRD][4CRSRR]INT.[SPC]
DATOS[CRSRD]"
130 INPUT"[2CRSRD]CUANTOS[SPC]NOMBRE
S";M
140 DIMA$(M)
150 L=M
160 FORN=1TOL
170 INPUT"[CRSRD]NOMBRE";A$(N)
180 NEXT
190 PRINT"[2CRSRD][3CRSRR]PULSE[SPC]
UNA[SPC]TECLA"
200 GETE$:IFE$=""THEN200
210 PRINT"[CLR]"
220 PRINT"[3CRSRD][4CRSRR]DIRECCIONE
S[SPC]"
230 DIMD$(M)
240 FORN=1TOL
250 PRINT"[2CRSRD]"A$(N):INPUT"[CRSRD]
DIRECCION";D$(N)
260 NEXT
270 PRINT"[2CRSRD][3CRSRR]PULSE[SPC]
UNA[SPC]TECLA"
280 GETE$:IFE$=""THEN280
290 PRINT"[CLR][3CRSRD][4CRSRR]TELEF
ONOS"
300 DIMT$(M)
310 FORN=1TOL
320 PRINT"[2CRSRD]"A$(N):INPUT"[CRSRD]
TELEFONO";T$(N)
330 NEXT
340 PRINT"[2CRSRD][3CRSRR]PULSE[SPC]
UNA[SPC]TECLA"
350 GETE$:IFE$=""THEN350
360 PRINT"[CLR][3CRSRD][6CRSRR]RELAC
ION"
370 DIMR$(M)
380 FORN=1TOL
390 PRINT"[2CRSRD]"A$(N):INPUT"[CRSRD]
RELACION";R$(N):NEXT
400 PRINT"[2CRSRD][3CRSRR]PULSE[SPC]
UNA[SPC]TECLA"
410 GETE$:IFE$=""THEN410
420 PRINT"[CLR][3CRSRD][8CRSRR]OTROS
"
430 DIMOT$(M)
440 FORN=1TOL
450 PRINT"[2CRSRD]"A$(N):INPUT"[CRSRD]
OTROS";OT$(N):NEXT
460 PRINT"[2CRSRD][3CRSRR]PULSE[SPC]
UNA[SPC]TECLA"
470 GETE$:IFE$=""THEN470
480 GOTO740
490 PRINT"[CLR]":FORN=1TOL:PRINT"[CRSRD]
[2CRSRR]-"A$(N):PRINT"[2CRSRR]-"D$(N
):PRINT"[2CRSRR]-"T$(N)
500 PRINT"[2CRSRR]-"R$(N):PRINT"[2CRSRR]
-"OT$(N)
510 NEXT
520 PRINT"[2CRSRD][3CRSRR]PULSE[SPC]
UNA[SPC]TECLA"
530 GETE$:IFE$=""THEN530
540 GOTO740
550 REM SUBRRUTINA DE CLASIFICACION
560 PRINT"[CLR][CRSRD][4CRSRR]CLASIF
ICACION[2CRSRD]"
570 FORN=1TOL:PRINT"[BLK][CRSRD]NUME
RO"N:PRINT"[WHT]"A$(N):NEXT
580 PRINT"[2CRSRD][3CRSRR]PULSE[SPC]
UNA[SPC]TECLA"
590 GETE$:IFE$=""THEN590
600 GOTO740
610 REM SUBRRUTINA DE BUSQUEDA
620 PRINT"[CLR][2CRSRD][CRSRR]INTROD
UCE[SPC]EL[SPC]NUMERO"
630 PRINT"[CRSRD][CRSRR]DE[SPC]LA[SPC]
PERSONA[SPC]DESEADA"
640 INPUT"[2CRSRD]NUMERO";H
650 IFH>LTHEN640
660 PRINT"[2CRSRD]LA[SPC]PERSONA[SPC]
DESEADA[SPC]ES"
670 PRINT"[CRSRD][2CRSRR]-"A$(H)
680 PRINT"[2CRSRR]-"D$(H)
690 PRINT"[2CRSRR]-"T$(H)
700 PRINT"[2CRSRR]-"R$(H)
710 PRINT"[2CRSRR]-"OT$(H)
720 PRINT"[2CRSRD][3CRSRR]PULSE[SPC]
UNA[SPC]TECLA"
730 GETE$:IFE$=""THEN730
740 POKE37158,28:POKE37159,0:PRINT"[CLR]
[BLK][CRSRD][4CRSRR]BASE[SPC]DE[SPC]
DATOS"
750 PRINT"[4CRSRR][4COMMU][SHIFT SPC]
[2COMMU][SHIFT SPC][5COMMU]"
760 PRINT"[CRSRD][CRSRR]1-INTRODUCIR
[SPC]DATOS"
770 PRINT"[CRSRD][CRSRR]2-PRESENTAR[SPC]
DATOS"
780 PRINT"[CRSRD][CRSRR]3-CLASIFICAR
[SPC]DATOS"
790 PRINT"[CRSRD][CRSRR]4-BUSQUEDA[SPC]
DE[SPC]DATO"
800 PRINT"[CRSRD][CRSRR]5-SALVAR[SPC]
DATOS"
810 PRINT"[CRSRD][CRSRR]6-CARGAR[SPC]
DATOS"
820 PRINT"[CRSRD][CRSRR]7-BORRAR[SPC]
DATOS"
830 PRINT"[CRSRD][CRSRR]8-CAMBIAR[SPC]
DATO"
840 GETQ$:IFQ$="1"THENGOTO110
850 IFQ$="2"THENGOTO490
860 IFQ$="3"THENGOTO550
870 IFQ$="4"THENGOTO610
880 IFQ$="5"THENGOTO940
890 IFQ$="6"THENGOTO1080
900 IFQ$="7"THENCLR:RUN
910 IFQ$="8"THENGOTO1210
920 IFQ$=""THEN840
930 GOTO840
940 POKE37158,38:POKE37159,72
950 INPUT"[CLR][2CRSRD]NOMBRE[SPC]FI
CHERO";GG$
960 INPUT"[2CRSRD]CUANTOS[SPC]DATOS"
;J
970 OPEN1,1,1,GG$
980 FORN=1TOJ
990 FORO=1TO5
1000 PRINT#1,A$(N)
1010 PRINT#1,D$(N)
1020 PRINT#1,T$(N)
1030 PRINT#1,R$(N)
1040 PRINT#1,OT$(N)
1050 NEXT:NEXT
1060 CLOSE1
1070 GOTO740
1080 POKE37158,38:POKE37159,72:INPUT
"[CLR][2CRSRD]NOMBRE[SPC]FICHERO";GG
$:INPUT"[2CRSRD]CUANTOS[SPC]DATOS";J

1090 OPEN1,1,0,GG$
1100 FORN=1TOJ
```

```
1110 FORO=1TO5
1120 INPUT#1,A$(N)
1130 INPUT#1,D$(N)
1140 INPUT#1,T$(N)
1150 INPUT#1,R$(N)
1160 INPUT#1,OT$(N)
1170 NEXT:NEXT
1180 CLOSE1
1190 L=J
1200 GOTO740
1210 POKE37158,38:POKE37159,72:PRINT
"[CLR][3CRSRD][3CRSRR]QUE[SPC]NUMERO
[SPC]HAY"
1220 PRINT"[CRSRD][3CRSRR]QUE[SPC]VA
RIAR"
```

```
1230 INPUTZ
1240 PRINT"[2CRSRD][2CRSRR]-"A$(Z)
1250 PRINT"[2CRSRR]-"D$(Z)
1260 PRINT"[2CRSRR]-"T$(Z)
1270 PRINT"[2CRSRR]-"R$(Z)
1280 PRINT"[2CRSRR]-"OT$(Z)
1290 PRINT"[3CRSRD]":INPUT"NOMBRE";A
$(Z)
1300 INPUT"[CRSRD]DIRECCION";D$(Z)
1310 INPUT"[CRSRD]TELEFONO";T$(Z)
1320 INPUT"[CRSRD]RELACION";R$(Z)
1330 INPUT"[CRSRD]OTROS";OT$(Z)
1340 N=Z
1350 GOTO740
```

C-64

Assembler

Miguel Jaque
C/ Francisca Moreno, 3. Telf. 43 21 23
28001 Madrid

Este programa permite programar directamente en el código nemotécnico del 6510. La notación que utiliza no es la correcta, sino parecida. Esto es porque utilizo la longitud de la cadena para hallar el modo de direccionamiento.

Es sólo ensamblador, no desensamblador. Es decir, que sólo ves lo que has programado en lo que queda en la pantalla. No puedes sacar un listado. Pero aún así es un buen complemento para programas como el arriba periscopio.

En primer lugar el programa pide la dirección de memoria en decimal en la cual se va a iniciar el programa. En el margen izquierdo irá indicando la dirección alcanzada.

Los nemotécnicos deben escribirse de este modo (sin espacios).

Direccionamiento.

Implícito	NME
Inmediato	NME#FF
Pág. cero	NME$FF
Pág. cero x	NME$FFx
Absoluto	NME$FFAA (Byte Alto, Byte Bajo)
Absoluto x	NME$(FFAA)x " "
Absoluto y	NME$(FFAA)y " "
Indirecto x	NME$(FF)x " "
Indirecto y	NME$(FF)y " "
Salto indirecto	JMP$(FFAA) " "

En los saltos condicionales (BEQ,BNE...) debe expresarse BNE#03. Siendo el número de bytes a saltar.

Por último, si en vez de poner una instrucción ponemos una x, el programa parará.

ASSEMBLER

```
1 INPUT"DIR";D
2 PRINTD;:INPUTI$:IFI$="X"THENEND
3 ON LEN(I$)GOSUB22,22,5,22,22,6,30,
10,12,49,17,22
4 GOTO2
5 B$=I$:Q=1:GOSUB33 :POKED,R:D=D+1:R
ETURN
6 IF MID$(I$,4,1)="#"THENQ=2:GOTO8
7 Q=3
8 A$=RIGHT$(I$,2):GOSUB23:B$=LEFT$(I
$,3):GOSUB33
9 POKED,R:POKED+1,A:D=D+2:RETURN
10 B$=LEFT$(I$,3):Q=5:A$=MID$(I$,5,2
):GOSUB23:B=A:A$=RIGHT$(I$,2):GOSUB2
3
11 GOSUB33: POKED,R:POKED+1,A:POKED+
2,B:D=D+3:RETURN
12 IF RIGHT$(I$,1)="X"THENQ=8:GOTO14

13 Q=9
14 A$=MID$(I$,6,2):GOSUB23
15 B$=LEFT$(I$,3):GOSUB33
16 POKED,R:POKED+1,A:D=D+2:RETURN
17 IFRIGHT$(I$,1)="X"THENQ=6:GOTO19
18 Q=7
19 A$=MID$(I$,6,2):GOSUB23:B=A:A$=MI
D$(I$,8,2):GOSUB23:B$=LEFT$(I$,3)
20 GOSUB33
21 POKED,R:POKED+1,A:POKED+2,B:D=D+3
:RETURN
22 PRINT"ERR.":RETURN
23 Z$=LEFT$(A$,1):IFASC(Z$)>64THENF=
ASC (Z$)-55:GOTO25
24 A=VAL(Z$)*16:GOTO26
25 A=F*16
26 Z$=RIGHT$(A$,1):IFASC(Z$)>64THEN
F=ASC (Z$)-55:GOTO28
27 A=A+VAL(Z$):GOTO29
28 A=A+F
29 RETURN
30 B$=LEFT$(I$,3):Q=4:GOSUB33
31 A$=MID$(I$,5,2):GOSUB23
32 POKED,R:POKED+1,A:D=D+2:RETURN
33 READJ$:IFJ$=B$THEN35
34 GOTO33
35 FORF=1TOQ:READR:NEXT:RESTORE:RETU
RN
36 DATAADC,,105,101,117,109,125,121,
97,113,AND,,41,37,53,45,61,57,33,49,
ASL
37 DATA,10,,6,22,14,30,BCC,,144,BCS,
,176,BEQ,,240,BIT,,,36,,44,BMI
38 DATA,48,BNE,,208,BPL,,16,BRK,0,BV
C,,80,BVS,,112,CLC,24,CLD
39 DATA216,CLI,88,CLV,184,CMP,,201,1
97,213,205,221,217,193,209,CPX,,224,
228,,
40 DATA236,CPY,,192,196,,204,DEC,,,1
98,214,206,222,DEX,202,DEY,136,EOR22
8,,
41 DATA,73,69,85,77,93,89,65,81,INC,
,,230,246,238,254,INX,232,INY,260,JM
P
42 DATA,,,,76,JSR,,,,,32,LDA,,169,16
5,181,173,189,185,161,177,LDX
43 DATA,162,166,182,174,,190,LDY,,16
0,164,180,172,188,LSR,76,,70,86,78,9
4
44 DATANOP,234,ORA,,9,5,21,13,29,25,
1,17,PHA,72,PHP,8,PLA,104,PLP,404
45 DATAROL,42,,38,54,46,62,ROR,106,,
102,118,110,126,RTI,64,RTS,96,SBC,,2
33
46 DATA229,245,257,253,249,225,241,S
EC,56,SED,248,SEI,120,STA,,,133,149,
141
47 DATA157,153,129,145,STX,,,134,150
,142,STY,,,132,148,140,TAX,170,TAY,1
68
48 DATATSX,186,TXA,138,TXS,154,TYA,1
52
49 A$=MID$(I$,6,2):GOSUB23:B=A:A$=MI
D$(I$,8,2):GOSUB23
50 POKED,108:POKED+1,A:POKED+2,B:D=D
+3:RETURN
```

VIC-20

Baloncesto y Estadísticas

René Rodríguez
Rbla. Just Oliveras, 66. 2.ª-1.ª
Telf.: 337 43 99
L'Hospitalet (Barcelona)

Con gusto me dirijo a ustedes para ofrecerles cinco de mis programas, con la esperanza tengan a bien publicarme al menos uno de ellos. Tres son juegos (en este aspecto no soy muy original), pero los dos restantes son educativos y muestran perfectamente las utilidades a las que se puede llegar con el VIC-20, y que es capaz de algo más que acribillar marcianitos.

Baloncesto

Cuando la selección española de baloncesto consiguió aquella medalla de plata en Los Angeles, en un momento de éxtasis triunfal confeccioné este programa. El jugador se enfrenta a los temibles Victroters, dirigidos por el banquillo por el famoso entrenador computerizado VIC-20. Un consejo: no se desanimen tras jugar la 1.ª partida, ni la 2.ª, ni la 3.ª, ni la 4.ª...

Las reglas del juego son complicadas, pero todo juego emocionante entraña dificultad. Un amigo mío consiguió perder por sólo ocho puntos tras jugar... ¡12 partidas! El partido consta de 8 minutos y el tiempo de posesión de balón es de 20 segundos. Como los jugadores de los Victroters se mueven aleatoriamente, el jugador se ve obligado a efectuar un pase, y el ordenador no. (algo parecido ocurría en "Batalla Naval").

Estadísticas

El último programa, "Estadísticas", es muy útil para realizar gráficos de porcentajes en Geografía para 2º de BUP.

Su presentación tal vez no es muy buena, pero la razón es que me he topado con las limitaciones de las exiguas 3,5K del VIC.

El menú te ofrece cuatro posibilidades de trabajo:

1) Representar una pirámide de edades.
2) Representar datos en columnas proporcionales horizontales (histogramas).
3) Representar la pluviosidad anual de una ciudad (climograma).
4) Cargar datos de la cassette previamente grabados en ella. Creo que es muy interesante, ya que permite asimismo la clasificación de datos al grabarlos en cinta. Además lo veo original ya que no sé de un programa parecido para el VIC sin ampliar.

Y esto es todo. Espero volver a escribirles muy pronto para enseñarles mis programas. Yo soy un suscriptor de su estupenda revista que no me he atravido a escribirles antes tan sólo por temor. Pero creo que todos debemos participar y ayudarnos en la tan divertida tarea de programar.

BALONCESTO

```
10 PRINT"[CLR]":TI$="000000":GOTO800

20 POKEJ+M,32:IFA$="[F1]"THENMM=-22:
RETURN
30 IFA$="[F3]"THENMM=22:RETURN
40 IFA$="4"THENMM=-1:RETURN
50 IFA$="5"THENMM=1:RETURN
70 IFA$="[SPC]"ANDE=1THENPOKEJ,37:J=
J+M:GOSUB200:POKEJ,32:MM=-1*MM
80 RETURN
```

```
100 PJ=PEEK(J):PM=PEEK(J+M):IFPJ=32A
NDPM=32THENRETURN
110 IFPJ=41ORPM=41THENPRINT"[HOM]FUE
RA":GOTO400
120 IFPM=42THENPRINT"[HOM]FUERA":GOT
O400
125 IFPJ=37ANDE=2ANDINT(RND(1)*3)=1T
HENPRINT"[HOM]PELOTA[SPC]PERDIDA":GO
TO400
130 IFPJ=44THENPRINT"[HOM]PELOTA[SPC]
PERDIDA":GOTO400
140 IFPM=44THENPRINT"[HOM]PELOTA[SPC]
PERDIDA":GOTO400
150 IFPM=37+WTHENPRINT"[HOM]COLISION
":GOTO400
155 IFPM=35ANDPR=1THENPRINT"[HOM]OBL
IGADO[SPC]HACER[SPC]1[SPC]PASE":GOTO
400
160 IFPM=35THENPRINT"[HOM]CANASTA":G
OTO460
170 RETURN
200 POKEJ,32:J=J+M:IFPEEK(J)=32THENP
OKEJ,36:FORA=1TO40:NEXT:GOTO200
205 IFPEEK(J)=37THENGOSUB300:PR=0:RE
TURN
210 IFPEEK(J)=44ANDE=1THENGOSUB400:G
OTO915
215 IFPEEK(J)=44ANDE=2THENV=7:GOSUB3
00:M=-1*M:RETURN
220 IFPEEK(J)=41ORPEEK(J)=42THENPRIN
T"[HOM]FUERA":GOSUB400:GOTO915
225 IFPEEK(J)=35ANDPR=1THENPRINT"[HOM]
OBLIGADO[SPC]PASAR":GOSUB400:GOTO915

230 IFPEEK(J)=35THENPRINT"[HOM]CANAS
TA":GOTO460
240 GOTO200
300 FORA=1TO3:IFPEEK(J-24+A)=44-VTHE
NK=J-24+A:RETURN
310 NEXT:FORA=1TO3:IFPEEK(J+20+A)=44
-VTHENK=J+20+A:RETURN
320 NEXT:IFPEEK(J+1)=44-VTHENK=J+1:R
ETURN
330 IFPEEK(J-1)=44-VTHENK=J-1
340 RETURN
400 FORA=1TO4000:NEXTA:U=-1*U:E=E+U:
POKE7168,CA:POKE7169,PU:GOTO915
440 PRINTB$:PRINT"[BLK][HOM]"TAB(18)
C$:PRINT"[9CRSRD]VICTROTERS";"[3CRSRR]
BARSA":PRINT""TAB(7)PU;
441 RETURN
460 U=-1*U:E=E+U
463 POKE36874,0:FORL=1TO7:FORH=200TO
(220+L*2):POKE36876,H:NEXTH:NEXTL:PO
KE36876,0
465 IFE=2THENCA=CA+2:GOTO480
470 PU=PU+2
480 POKE7168,CA:POKE7169,PU:GOTO915
800 POKE36879,220:PRINT"[4CRSRD][BLK]
[6CRSRR][RVSON]BASKETVIC[RVSOFF]"
810 PRINT"[WHT][2CRSRD][6CRSRR]AGOST
O-84":PRINT"[BLK][2CRSRD][2CRSRR]F1[SPC]
ARRIBA[2SPC]F3[SPC]ABAJO"
820 PRINT"[BLU][2CRSRD]4=IZQUIERDA[SPC]
5=DERECHA":PRINT"[CRSRD][2CRSRR]SPAC
E=PASE[SPC]Y[SPC]TIRO"
830 WAIT198,1
900 POKE52,28:POKE56,28:CLR:FORA=716
8TO7679:POKEA,PEEK(A+25600):NEXT:POK
E36869,255
907 FORA=1TO8:READB:NEXTA
910 FORA=7448TO7527:READB:POKEA,B:NE
XT:E=1:U=-1:POKE7168,0:POKE7169,0:PO
KE36878,15
915 PR=0:POKE198,0:IFE=1THENCLR:PR=1
:E=1:U=-1:CA=PEEK(7168):PU=PEEK(7169
):GOTO917
916 W=12
917 J=7910:D=30720:K=7909:M=1:B$="[BLK]
[HOM][CRSRR][7CRSRD]+++&[CRSRD]&[CRSRD]
[CRSRL]'[CRSRD][CRSRL]([CRSRD][2CRSRL]
([2CRSRL][CRSRD]+[2CRSRL]+[2CRSRL]+[3CRSRU]
#":PRINT"[CLR]"
920 FORA=7724TO7745:POKEA,41:POKEA+1
4*22,41:NEXT:FORA=7724TO8032STEP22:P
OKEA,42
925 POKEA+21,42:NEXT
930 RESTORE:FORA=1TO4:READB:POKEW+B,
44:POKEB+D+W,2:NEXT:FORA=1TO4:READB:
POKEW+B,37
940 POKEW+B+D,6:NEXT
992 C$="[7CRSRD]+++[4CRSRL]([2CRSRL]
[CRSRD]([2CRSRL][CRSRD]'[CRSRD]&[CRSRD]
&[CRSRD]+++[2CRSRL][3CRSRU]#":POKEK,
44
999 L=20:ONEGOTO1000,2000
1000 L=L-.23:IFVAL(TI$)=>800THEN3000

1002 GETA$:IFA$<>""THENPOKEJ,32:POKE
J+M,32:GOSUB20:M=MM:GOTO1005
1003 POKEJ,32:POKEJ+M,32
1005 J=J+M:GOSUB100
1010 POKEJ,37:POKEJ+D,6:POKEJ+M,36:P
OKE36874,0
1015 IFINT(RND(1)*19)=0THEN1040
1020 POKEK,32:K=K+M:IFPEEK(K)=37THEN
PRINT"[HOM]PERSONAL":FORA=1TO4000:NE
XTA:GOTO915
1023 IFPEEK(K)=44THENK=K+1
1025 POKEK,44:POKEK+D,2
1030 IFL<1THENPRINT"[HOM]TIEMPO[SPC]
AGOTADO":GOSUB400:GOTO915
```

```
1040 GOSUB440:PRINTTAB(14)CA:POKE368
74,200:PRINTTI$;:PRINTINT(L):GOTO100
0
2000 L=L-.23:IFVAL(TI$)=>800THEN3000

2002 FORA=1TO5:GETA$:IFA$<>""THENGOS
UB20:POKEK,32:K=K+MM:POKEK,37:POKEK+
D,6
2010 NEXT
2020 IFM<>1ANDPEEK(J+2)=32THENPOKEJ+
M,32:M=1:GOTO2060
2030 IFINT(1+RND(1)*2)=2THENPOKEJ+M,
32:M=22:GOTO2060
2050 POKEJ+M,32:M=-22
2060 POKEJ,32:J=J+M:POKE36874,0
2070 GOSUB100:FORA=1TO8:IFPEEK(J+M*A
)=35THENPOKEJ,44:POKEJ+D,2:J=J+M:GOS
UB200
2072 IFPEEK(J+M*A)=44THENPOKEJ,44:PO
KEJ+D,2:J=J+M:GOSUB200:GOTO2080
2075 NEXT:POKE36874,200
2080 POKEJ,44:POKEJ+D,2:POKEJ+M,36:G
OSUB440;:PRINT""TAB(14);CA:PRINTTI$;

2090 PRINTINT(L):GOTO2000
3000 PRINT"[HOM][4CRSRD]FINAL[SPC]DE
L[SPC]PARTIDO";PU;CA:POKE36878,0
3005 FORA=1TO4000:NEXT:POKE198,0
3010 WAIT198,1:RUN
6999 DATA7769,7794,7949,7968,7770,77
95,7971,7990
7000 DATA0,0,24,36,36,24,0,0,0,0,0,2
4,24,0,0,0,24,24,0,126,90,90,24,60,1
28,64,32,16,8
7010 DATA4,2,1,1,1,1,1,1,1,1,1,1,2,4
,8,16,32,64,128,0,0,0,255,255,0,0,0,
24,24,24,24,24
7020 DATA24,24,24,255,0,0,0,0,0.0,0,
24,24,0,126,90,90,24,60
```

ESTADISTICAS VIC-20

```
40 PRINT"[CLR][BLU][RVSON]MENU[RVSOFF]
":DI=30720:FI=7705:CO=1
50 PRINT"[4CRSRD][RED]1)[SPC]PIRAMID
E[SPC]EDADES":PRINT"[2CRSRD]2)[SPC]T
ASAS[SPC]VARIAS"
55 PRINT"[2CRSRD]3)[SPC]CLIMOGRAMA":
PRINT"[2CRSRD]4)[SPC]LECTURA[SPC]DE[SPC]
FICHEROS":PRINT"[2CRSRD][BLU]JURADO[SPC]
10-84"
60 GETA$:IFA$=""ORVAL(A$)<1ORVAL(A$)
>4THEN60
70 ONVAL(A$)GOTO100,200,300,400
100 INPUT"[CLR][CRSRD][RED]PAIS",P$:
DIMH(20):DIMM(20)
110 FORA=1TO20:PRINT"[CRSRD][BLU][SPC]
%[SPC]",(A-1)*5;"-",A*5;"[SPC]HOMBRE
S":INPUTH(A)
120 PRINT"[CRSRD][RED][SPC]%[SPC]";(
A-1)*5;"-";A*5;"[SPC]MUJERES":INPUTM
(A):NEXT
130 PRINT"[CLR]":FORA=19TO0STEP-1:PR
INT"[BLK][9CRSRR]";A*5:NEXT:PRINT"%9
87654321[2SPC]123456789%[CRSRL]"
140 PRINT"[CRSRU][RED]";P$
150 CO=0:FI=8109:FORA=1TO20:FORX=1TO
M(A):POKEFI+X,160:POKEFI+X+DI,0:NEXT

160 RE=(M(A)-INT(M(A)))*10:GOSUB501:
FI=8109-22*A:NEXT:FI=8108:R=2:S=10:T
=1:U=128
170 FORA=1TO20:FORX=1TOH(A):POKEFI-X
,160:POKEFI-X+DI,0:NEXT:RE=(H(A)-INT
(H(A)))*10
180 GOSUB501:FI=8108-22*A:NEXT
190 WAIT198,1:POKE198,0:PRINT"[CRSRD]
DESEA[SPC]ABRIR[SPC]FICHERO?"
191 GETA$:IFA$=""THEN191
192 IFA$="N"THENRUN
193 GOTO800
200 INPUT"[CLR][3CRSRD]NUMERO[SPC]DE
[SPC]ENTES";N
210 IFN>21THENPRINT"[2CRSRD][RVSON]N
UMERO[SPC]EXCESIVO[RVSOFF]":FORA=1TO
3000:NEXT:GOTO200
220 DIMB$(N):DIMC(N):FORB=1TON:PRINT
"[CRSRD][RED]ABREVIATURA[SPC]ENTE";B
:INPUTB$(B)
230 PRINT"[BLU][RVSON]DATO[SPC]DE[SPC]
[RVSOFF]",B$(B):INPUTC(B):NEXT
240 INPUT"[2CRSRD][BLK]ESCALA";ES
250 FORB=1TON:IFC(B)/ES>18THENPRINT"
[RVSON][RED][2CRSRD]ESCALA[SPC]INDEB
IDA[RVSOFF]":GOTO240
260 NEXT:PRINT"[CLR]":FORB=1TON:CO=C
O+1:IFCO=7THENCO=2
270 PRINTB$(B):D=INT(C(B)/ES):FORX=1
TOD:POKEFI+X,160
280 POKEFI+30720+X,CO:NEXT:RE=(C(B)/
ES-D)*10:X=X+1:GOSUB500:FI=FI+22:NEX
T
290 PRINTNO$:WAIT198,1:POKE198,0:PRI
NT"[CRSRD]DESEA[SPC]ABRIR[SPC]FICHER
O?"
291 GETA$:IFA$=""THEN291
292 IFA$="N"THENRUN
293 GOTO750
300 DIMPL(12):DIMB$(12):PRINT"[CLR]"

310 FORA=1TO12:READB$(A):PRINT"[BLU]
[CRSRD]PLUVIOSIDAD[SPC]";"[RED]";B$(
A);"[BLK]":INPUTPL(A):NEXT
315 INPUT"[CRSRD]LUGAR";L$
320 INPUT"[2CRSRD][BLU]ESCALA";ES
330 FORA=1TO12:IFPL(A)/ES>17THENPRIN
T"[RED][RVSON][CRSRD]ESCALA[SPC]INDE
BIDA":GOTO320
335 NEXT
340 PRINT"[CLR][18CRSRD][8CRSRR]EFMA
MJJASOND[2CRSRR][RED]";L$;"[CRSRU]":
FORA=0TO17:PRINT"[2CRSRU][BLK]";
350 PRINTA*ES:NEXT
360 SI=8062:FORA=1TO12:D=-(INT(PL(A)
/ES)-1)*22:FORX=0TODSTEP-22:POKESI+X
,160
370 POKESI+X+30720,0:NEXT:RE=(PL(A)/
ES-(-D/22+1))*10:GOSUB600:SI=SI+1:NE
XT
390 WAIT198,1:POKE198,0:PRINT"[CRSRD]
DESEA[SPC]ABRIR[SPC]FICHERO?"
391 GETA$:IFA$=""THEN391
392 IFA$="N"THENRUN
393 GOTO700
400 INPUT"[CLR][BLK][RVSON]NOMBRE[RVSOFF]
";NO$:PRINT"[2CRSRD][BLU]PARA[SPC]QU
E[SPC]OPCION[SPC]DESEA[SPC][CRSRD]LE
ER[SPC]DATOS[SPC](1-3)"
410 GETA$:IFA$=""ORVAL(A$)<1ORVAL(A$
)>3THEN410
420 PRINT"[CLR][CRSRD][RED][RVSON]CA
RGANDO[SPC]DATOS":ONVAL(A$)GOTO430,7
70,730
430 DIMH(20):DIMM(20):OPEN1,1,0,NO$:
INPUT#1,P$
450 FORA=1TO20:PRINT"[CRSRD][BLU]EDA
D",(A-1)*5;"-";A*5;"[SPC]HOMBRES":IN
PUT#1,H(A):PRINTH(A)
460 PRINT"[CRSRD][RED]EDAD[SPC]";(A-
1)*5;"[SPC]MUJERES":INPUT#1,M(A):PRI
NTM(A):NEXT
470 CLOSE1:GOTO130
```

```
500 IFC(B)=0THENPO=32:X=X-1:GOTO570
501 IFRE=0THENRETURN
502 IFH(A)<1ANDR=2THENX=X-1:GOTO505
503 IFM(A)<1ANDR=0THENX=X-1
505 IFRE<2THENPO=101+R:GOTO570
510 IFRE<3THENPO=116-S:GOTO570
520 IFRE<4THENPO=117+T:GOTO570
530 IFRE<6THENPO=97+U:GOTO570
540 IFRE<7THENPO=246-T:GOTO570
550 IFRE<8THENPO=234+S:GOTO570
560 PO=231-R
570 IFD<1ANDH(5)=0THENX=X-1
575 IFR=2THENPOKEFI-X,PO:POKEFI-X+DI
,CO:RETURN
580 POKEFI+X,PO:POKEFI+X+DI,CO:RETUR
N
600 IFPL(A)=0THENPO=32:GOTO670
605 IFRE=0THENRETURN
610 IFRE<2THENPO=100:GOTO670
620 IFRE<3THENPO=111:GOTO670
630 IFRE<4THENPO=121:GOTO670
640 IFRE<6THENPO=98:GOTO670
650 IFRE<7THENPO=248:GOTO670
660 IFRE<8THENPO=247:GOTO670
665 PO=227
670 IFPL(A)<ESTHENX=X+22
680 POKESI+X,PO:POKESI+X+DI,0:RETURN

700 PRINT"[CLR][2CRSRD]GRABANDO[SPC]
DATOS":INPUT"[2CRSRD][BLK]NOMBRE";NO
$:OPEN1,1,1,NO$:FORA=1TO12:PRINT#1,B
$(A)
710 PRINT#1,PL(A):NEXT:PRINT#1,L$:PR
INT#1,ES
720 CLOSE1:RUN
730 DIMB$(12):DIMPL(12)
735 PRINT"[CLR][2CRSRD][BLK][RVSON]L
EYENDO":OPEN1,1,0,NO$:FORA=1TO12:INP
UT#1,B$(A):INPUT#1,PL(A)
740 PRINTB$(A):PRINTPL(A):NEXT:INPUT
#1,L$:INPUT#1,ES:GOTO340
750 INPUT"[CRSRD]NOMBRE";N$:OPEN1,1,
1,N$:PRINT#1,N:PRINT#1,ES:FORA=1TON
760 PRINT#1,B$(A):PRINT#1,C(A):NEXT:
CLOSE1:RUN
770 OPEN1,1,0,NO$:INPUT#1,N:DIMB$(N)
:DIMC(N):INPUT#1,ES:FORA=1TON
780 INPUT#1,B$(A):INPUT#1,C(A):PRINT
B$(A);C(A):NEXT:A=0:GOTO250
800 PRINT"[CLR]GRABANDO":INPUT"[CRSRD]
NOMBRE";N$:OPEN1,1,1,N$:PRINT#1,P$:F
ORA=1TO20:PRINT#1,H(A)
810 PRINT#1,M(A):NEXT:CLOSE1:RUN
2000 DATAENERO,FEBRERO,MARZO,ABRIL,M
AYO,JUNIO,JULIO,AGOSTO,SETIEMBRE,OCT
UBRE,NOVIEMBRE
2010 DATADICIEMBRE
```

C-64

Proof-Sprite

Juan Jesús Daryanani Hormiga
Suscriptor 5676
C/ San Martín, 64. 2-5
S. C. de Tenerife

El programa que os adjunto en la cinta de cassette no necesita explicación alguna, ya que desde que se carga y se ejecuta, te explica hasta el último detalle. Mi idea era mandar el programa protegido, pero mis conocimientos no son tantos, por lo que me es casi imposible. Os rogaría que si sabéis la forma de proteger un programa en una cinta, me la digáis.

Tengo en proyecto un par de programas más, así que cuando los acabe me acordaré de vosotros y os los enviaré.

Nota: Cuando nos mandéis los programas, no debéis mandarlos protegidos, porque en tal caso no podríamos listarlos.

PROOF-SPRITE

```
4 FORN=0TO62:POKE832+N,0:NEXT
5 PRINT"[CLR]":POKE53280,2:POKE53281
,14:POKE646,1
10 FORA=1024TO1063:POKEA,170:POKEA+5
4272,7:NEXT
20 FORA=1984TO2023:POKEA,170:POKEA+5
4272,7:NEXT
30 FORA=1064TO1944STEP40:POKEA,170:P
OKEA+54272,7:NEXT
40 FORA=1063TO2023STEP40:POKEA,170:P
OKEA+51272,7:NEXT
50 PRINTTAB(8);"[CRSRD][BLU][RVSON]D
EMOSTRACION[SPC]DE[SPC]SPRITES:"
60 PRINTTAB(14);"[BLU][RVSON]PROOF[SPC]
SPRITE":PRINT
70 PRINTTAB(1);"ESTE[SPC]PROGRAMA[SPC]
TE[SPC]SERVIRA[SPC]PARA[SPC]PROBAR"
80 PRINTTAB(1);"CUALQUIER[SPC]SPRITE
```

```
[SPC]QUE[SPC]ENCUENTRES[SPC]EN[SPC]A
L-"
90 PRINTTAB(1);"GUNA[SPC]REVISTA,YA[SPC]
QUE[SPC]TE[SPC]PERMITE[SPC]INTRO-[SPC]
"
100 PRINTTAB(1);"DUCIR[SPC]LOS[SPC]D
ATOS[SPC]QUE[SPC]APARECEN[SPC]EN[SPC]
LAS[2SPC]"
110 PRINTTAB(1);"LLAMADAS[SPC]LINEAS
[SPC]DATAS.[SPC]APROVECHANDO[2SPC]"
120 PRINTTAB(1);"ESTA[SPC]INTRODUCCI
ON[SPC]TAMBIEN[SPC]OS[SPC]ADJUNTO[SPC]
"
130 PRINTTAB(1);"UN[SPC]SPRITE[SPC]D
E[SPC]DEMOSTRACION[SPC]YA[SPC]QUE[SPC]
ME[2SPC]"
140 PRINTTAB(1);"HA[SPC]PARECIDO[SPC]
QUE[SPC]ES[SPC]EL[SPC]MEJOR[SPC]QUE[SPC]
HE[3SPC]"
150 PRINTTAB(1);"TENIDO[SPC]Y[SPC]QU
E[SPC]SIGO[SPC]TENIENDO.[SPC]DEBES[SPC]
SE-"
160 PRINTTAB(1);"GUIR[SPC]LAS[SPC]IN
STRUCCIONES[SPC]AL[SPC]PIE[SPC]DE[SPC]
LA[2SPC]"
170 PRINTTAB(1);"LETRA,SIN[SPC]EQUIV
OCARSE,Y[SPC]ANTE[SPC]TODO[SPC]NO[SPC]
"
180 PRINTTAB(1);"PULSES[SPC]LA[SPC]T
ECLA[SPC]RUN/STOP[SPC]YA[SPC]QUE[SPC]
SINO[SPC]"
190 PRINTTAB(1);"TODO[SPC]SE[SPC]IRI
A[SPC]A[SPC]LA[SPC]PORRA.Y[SPC]AHORA
[SPC]SIN[2SPC]"
200 PRINTTAB(1);"NADA[SPC]MAS[SPC],Y
[SPC],DESEANDO[SPC]QUE[SPC]ESTE[2SPC]
"
210 PRINTTAB(1);"PROGRAMA[SPC]SEA[SPC]
DE[SPC]TU[SPC]AGRADO,HASTA[SPC]LA[2SPC]
"
220 PRINTTAB(1);"PROXIMA(VEZ[SPC]QUE
[SPC]USES[SPC]ESTE[SPC]PROGRAMA).":P
RINT
230 PRINTTAB(6);"PULSA[SPC]ESPACIO[SPC]
PARA[SPC]CONTINUAR"
240 GETG$:IFG$=""THEN243
250 PRINT"[CLR]":CLR
255 FORN=0TO62:POKE832+N,0:NEXT
260 FORA=1024TO1063:POKEA,127:POKEA+
54272,7:NEXT
270 FORA=1984TO2023:POKEA,127:POKEA+
54272,7:NEXT
280 FORA=1024TO1984STEP40:POKEA,127:
POKEA+54272,7:NEXT
290 FORA=1063TO2023STEP40:POKEA,127:
POKEA+54272,7:NEXT:PRINT
300 PRINTTAB(12);"[BLU][RVSON]MENU[SPC]
DE[SPC]OPCIONES":PRINT:PRINT
310 PRINTTAB(5);"1.-[SPC]INTRODUCIR[SPC]
NUMEROS[SPC]BINARIOS"
320 PRINTTAB(5);"[CRSRD]2.-[SPC]INTR
ODUCIR[SPC]NUMEROS[SPC]DECIMALES"
330 GETO$:IFO$=""THEN330
340 ONVAL(O$)GOTO500,1000
350 IFO$<>"1"ORO$<>"2"THEN330
360 END
500 PRINT"[CLR]":POKE53280,14:POKE53
281,4:POKE646,1
510 PRINT:PRINTTAB(5);"[RVSON]INTORD
UCCION[SPC]DE[SPC]DATOS[SPC]BINARIOS
":PRINT:PRINT
520 DIMZ(70),ND(70):FORB=0TO62
530 PRINT"DATO[SPC]BINARIO[SPC]NUMER
O";B;"[SPC]";:INPUTA$
540 TL=0:C=0
550 FORX=8TO1STEP-1:C=C+1
560 TL=TL+VAL(MID$(A$,C,1))*2↑(X-1)
570 NEXTX
580 ND(B)=TL
590 NEXTB
600 PRINT"[CLR]":REM COMIENZO OF SPR
ITE
610 V=53248
620 POKEV+21,28
630 POKE2042,13:POKE2044,13:POKE2043
,13
640 FORN=0TO62:POKE832+N,ND(N):NEXT
650 POKEV+23,12:POKEV+29,12:REM
660 FORJ=1TO10:FORX=0TO200
670 POKEV+4,X:POKEV+8,X
680 POKEV+5,X:POKEV+9,100
690 NEXTJ:NEXTX
700 PRINT"[CLR]":POKE781,5:POKE782,2
4:PRINTTAB(16);"[YEL]PULSE[SPC]F1"
710 GETK$:IFK$=""THEN710
720 IFK$=CHR$(133)THENPOKE53280,2:PO
KE53281,14:GOTO255
730 IFK$<>CHR$(133)THEN710
740 END
1000 PRINT"[CLR]":PRINT:CLR
1010 PRINTTAB(4);"[RVSON]INTRODUCCIO
N[SPC]DE[SPC]DATOS[SPC]DECIMALES:":P
RINT:PRINT
1020 DIMNP(70):FORZ=0TO62
1030 PRINT"DATO[SPC]DECIMAL[SPC]NUME
RO";Z;"[SPC]";:INPUTNP(Z)
1040 NEXT
1050 V=53248:PRINT"[CLR]"
1060 POKEV+21,28:POKE2042,13:POKE204
3,13:POKE2044,13
1070 FORN=0TO62:POKE832+N,NP(N):NEXT

1080 POKEV+23,12:POKEV+29,12
1090 FORJ=1TO10:FORX=0TO200
1100 POKEV+4,X:POKEV+8,X:POKEV+5,X:P
OKEV+9,100
1110 NEXTX:NEXTJ:PRINT"[CLR]"
1120 POKE781,5:POKE782,24:PRINT TAB(
16);"[YEL]PULSA[SPC]F1"
1130 GETJ$:IFJ$=""THEN1130
1140 IFJ$=CHR$(133)THENPOKE53280,2:P
OKE53281,14:POKE646,1:GOTO255
1150 IFJ$<>CHR$(133)THEN1130
1160 END
```

VIC-20 SIN EXPANSION

Black Dog

René Rodríguez
Rbla. Just Oliveras, 66 2º 1ª
L'HOSPITALET (BARCELONA)
Telf. 337 43 99

Las características de este juego están explicadas en la primera parte del programa. Como está en catalán si queréis alguna aclaración sólo tenéis que decírmelo. Todas las explicaciones se pueden eliminar del programa, pero no os olvidéis de teclear la parte que sirve para crear los caracteres del juego. Antes de empezar a teclear la segunda parte no os olvidéis de grabar la primera. La segunda parte debe grabarse inmediatamente después de la primera. Cuando hagamos LOAD y RUN de la primera parte el VIC nos avisará para que entremos la segunda parte y podamos empezar el juego.

LISTADO 1

```
1 POKE36879,238:POKE36885,242
2 POKE52,28:POKE56,28
3 PRINT"[CLR][4CRSRD][3CRSRR][BLU]**
```

```
**************":PRINT"[3CRSRR]*[BLK]
[SHIFTR]ENE[SPC][SHIFTR]ODRIGUEZ[BLU]
*"
5 PRINT"[3CRSRR]*[5SPC][BLK]1984[5SPC]
[BLU]*"
6 PRINT"[BLU][3CRSRR]***************
*":PRINT"[3CRSRD][2CRSRR][SHIFTP]RES
ENTA:":FORT=1TO2000:NEXT
7 PRINT"[CRSRR][2CRSRD][CRSRR][WHT][4CRSRR]
[SHIFTB]LACK[SPC][SHIFTD]OG":FORT=1T
O3000:NEXT
10 PRINT"[CLR][CRSRD][2CRSRR][BLK][SHIFTE]
L[SPC]JOC[SPC]CONSISTEIX[SPC]ENFER[SPC]
ARRIBAR[SPC]A[SPC][SHIFTB]LACK[3SPC]
FINS[SPC]LA[SPC]SEVA[SPC]CASETA[SPC]
A[SPC]TRA"
15 PRINT"[CRSRU][3CRSRR]VES[SPC]DELS
[SPC]DIFERENTS[SPC]CIRCUITS."
20 PRINT"[2CRSRR][SHIFTE]L[SPC]BOTO[SPC]
BERNELL[SPC]SER-VEIX[SPC]PER[SPC]ANA
R[SPC]MES[SPC]LEN-TAMENT."
25 PRINT"[2CRSRR][SHIFTL]A[SPC]PUNTU
ACIO[SPC]VE[SPC]DO-[SPC]NADA[SPC]PER
[SPC]LA[SPC]DISTANCIA[SPC]RECORREGUD
A[SPC]I[SPC]PEL[5SPC]TEMPS."
30 PRINT"[2CRSRR][SHIFTA]]SPC]CADA[SPC]
PANTALLA[SPC]EL[2SPC]TEMPS[SPC]ESCOM
ENCA[SPC]DE[SPC]0.[SPC][SHIFTS]I[SPC]
ES[SPC]FA[SPC]TOT[SPC]EL[SPC]RECOR-"
35 PRINT"[CRSRU]REGUT[SPC]EN[SPC]MEN
YS[SPC]DE[SPC]45[2SPC]SEGONS,[SPC]ES
[CRSRR]GUANYEN[4SPC]PUNTS.[SPC][SHIFTS]
I[SPC]ES[SPC]TRIGA[SPC]MESES[SPC]RES
TEN."
40 PRINT"[CRSRD][CRSRR][SHIFTA]PRETA
[SPC]ESPAI"
45 GETA$:IFA$<>"[SPC]"THEN45
50 PRINT"[CLR][2CRSRR][CRSRD][SHIFTC]
ADA[SPC]250[SPC]O[SPC]300[SPC]PUNTSL
A[SPC]PANTALLA[SPC]ES[SPC]FA[SPC]MES
[SPC]DIFICIL."
55 PRINT"[2CRSRR][SHIFTE]L[SPC]NOMBR
E[SPC]DE[SPC]COPS[3SPC]QUE[SPC][SHIFTB]
LACK[SPC]POT[SPC]SER[SPC]ATRA-PAT[SPC]
ES[SPC]3."
60 PRINT"[CRSRD][3CRSRR][SHIFTB]ONA[SPC]
SORT!!!":FORT=1TO5000:NEXT:PRINT"[2CRSRD]
[CRSRR][SHIFTA]PRETA[SPC]ESPAI"
65 GETB$:IFB$<>"[SPC]"THEN65
70 PRINT"[CLR][CRSRD][CRSRR][SHIFTU]
N[SPC]MOMENT[SPC]QUE[SPC]CARREGO[SPC]
EL[SPC]PROGRAMA."
74 FORT=7168TO7680:POKET,PEEK(T+2560
0):NEXT
75 FORT=7344TO7343+10*8:READA:POKET,
A:NEXT
80 DATA126,102,66,90,94,126,255,0,40
,56,56,16,56,56,40,40
82 DATA64,194,252,60,60,38,50,2,64,1
94,252,60,60,102,195,0
84 DATA0,0,99,127,62,127,99,0,2,67,6
3,60,60,102,195,0
86 DATA2,67,63,60,60,100,76,64,0,0,0
,60,60,126,126,255
88 DATA126,102,66,66,66,66,255,0,0,2
55,168,216,168,80,32,0
95 PRINT"[CLR]":POKE198,7:POKE631,76
:POKE632,207:POKE633,159:POKE634,13
100 POKE635,82:POKE636,213:POKE637,1
3:POKE36885,240
```

LISTADO 2

```
1 POKE36879,238
2 POKE36878,15:H=27
3 Z=7680:I=36875:Y=1
5 POKE52,28:POKE56,28:POKE36869,255
9 PRINT"[CLR]"
10 FORN=0TO19
11 POKE7724+N,26
12 POKE7856+N,26
13 POKE7988+N,26
14 POKE8120+N,26
15 POKE8122+N,26
16 NEXTN
20 FORM=2TO21
21 POKE7790+M,26
22 POKE7922+M,26
23 POKE8054+M,26
24 NEXTM:FORT=1TO2:POKE7680+(18+T)*2
2-2,28+T:POKE38860-22*T,5*T-3:NEXT
25 PRINT"[20CRSRD]"
26 POKE37151,0:PA=37151:PB=37152:RB=
37154
28 FORT=1TO2000:NEXT:PRINT"[CRSRU][3CRSRR]
[BLK]CAMI[SPC]LLIURE!!!":POKE36875,2
25:FORT=1TO500:NEXT:Y=1
```

```
29 POKE36875,0:PRINT"[CRSRU][19SPC]"
:TI$="000000"
30 POKEZ+X+22*Y+30720,0:POKEZ+X+22*Y
,H
32 A=PEEK(PA):POKERB,127:B=PEEK(PB):
POKERB,255
36 PRINT"[CRSRU]"L;TAB(9)MID$(TI$,4,
1)":"MID$(TI$,5,2);TAB(18)3-V
38 IFTI$>="000045"THENPOKEI,190:IFTI
$>="000100"THENDY=0:TI$="000000":GOT
O120
45 IF(BAND128)=0THENX=X+1:DY=-1:L=L+
1:POKEI,190:H=27:GOTO50
46 IF(AAND16)=0THENX=X-1:DY=1:L=L+1:
POKEI,190:H=24:GOTO50
47 IF(AAND4)=0THENY=Y-1:DY=22:H=23:G
OTO50
48 IF(AAND8)=0THENY=Y+1:DY=-22:H=23
50 IF(AAND32)=0THENFORT=1TO50:NEXT
51 POKEI,0:IFINT(L/2)=L/2ANDH<>23THE
NH=28:IFDY=1THENH=25
55 IFPEEK(Z+X+22*Y)=26THEN120
60 IFX=-1ORX=22ORY=-1THEN120
65 IFX=20ANDY=19THENPOKE38838,2:POKE
38816,7:POKE8096,29:POKE8118,22:POKE
8117,32:GOTO135
70 POKEZ+X+22*Y+DY,32:GOTO30
120 POKEZ+X+22*Y+DY,31:POKEI,0
121 POKE36877,200:FORT=1TO100:NEXT:P
OKE36877,225:FORT=1TO100:NEXT:POKE36
877,0
123 V=V+1:IFV>=3THENPRINT"[CLR][4CRSRD]
[3CRSRR]HAS[SPC]FET"L"PUNTS":END
124 FORT=1TO500:NEXT
125 POKEZ+X+22*Y+DY,32:X=0:Y=1:DY=0
126 GOTO30
130 L1=-90:LL=L+L1:GOTO136
135 L1=7*(45-VAL(TI$)):LL=L+50+L1
136 FORJ=1TO5
140 POKE36876,200:FORT=1TO100:NEXT:P
OKE36876,220:FORT=1TO100:NEXT:POKE36
876,0
141 NEXTJ
144 PRINT"[CLR][3CRSRD][CRSRR]HAS[SPC]
FET"L+50"PUNTS":PRINT"[CRSRD][CRSRR]
I"L1"PER[SPC]TEMPS=[WHT]"LL:L=LL
145 FORT=1TO3500:NEXT
146 X=0:Y=1
147 IFL<300THEN9
148 PRINT"[CLR]"
150 FORG=0TO7
152 POKE7724+G,26:POKE7792+G,26:POKE
7856+G,26:POKE7924+G,26:POKE7988+G,2
6
155 POKE8056+G,26
156 NEXTG
160 POKE7868,26
165 POKE7933,26
166 FORS=14TO19
170 POKE8000,26
181 FORS=14TO19
182 POKE7724+S,26:POKE7792+S,26:POKE
7856+S,26:POKE7924+S,26:POKE7988+S,2
6
183 POKE8056+S,26
184 NEXTS
190 FORQ=0TO17
195 POKE7690+22*Q,26:POKE7737+22*Q,2
6
200 NEXTQ
205 FORR=0TO21
210 POKE8120+R,26
214 NEXTR
215 PRINT"[20CRSRD]"
218 IFL>500THENW=3
219 IFL>750THENW=7
220 IFL>1000THENW=24
221 FORJ=1TOW:READR:POKEZ+R,26:NEXT:
RESTORE
222 FORT=1TO2:POKE7680+(18+T)*22-2,2
8+T:POKE38860-22*T,5*T-3:NEXT
225 GOTO28
230 DATA121,56,385,73,90,349,368,205
,222,266,293,337
235 DATA104,85,354,305,161,134,148,1
73,217,236,280,365
```

SUPER-INTERESANTISIMO

PIRATERIA DE SOFT ...Y ALGO MAS

(Viene de pág. 57)

Para aligerar y simplificar los intercambios y contactos intercommodorianos, los que no deseen otro tipo de compra-venta, etc., enviadnos el cupón que, a partir de hoy, aparece en la revista en la misma sección de Marketclub, una inserción por cupón.

7) La carta que se refiere a RUN la hemos insertado en estas páginas porque, aunque no siendo lo mismo, tiene bastantes similitudes.

Nunca hemos hecho ningún secreto de que RUN americana, alemana, danesa, etc., sean de la misma compañía C.W. Communications (ver pág. 3). De hecho es un nombre que tenemos registrado internacionalmente. Muchos de estos artículos aparecen en las diversas nacionales en distintos idiomas. Los listados no son idénticos, varían según la adaptación hecha al idioma nacional. Antes no especificábamos los que eran traducidos, solamente firmábamos los nacionales. Desde el nº 8 de octubre del 84, lo ponemos en todos —autor, traductor y adaptador— para que quede bien claro su procedencia.

Conocemos Compute, Compute's Gazzete y bastantes otras, pero no, son nuestras y por lo tanto no tenemos derecho a utilizar su material.

Y aquí es donde aparece la simiralidad.

La mayoría de las revistas de informática, impresas o en cassette, tienen material traducido en mayores o menores cantidades, bien por derecho propio, como en nuestro caso, bien por copy-rights. Y hay algunas —sobre todo en forma de cassette— que no lo tienen pero se lo inventan. Al igual que se utilizan nombre y autorías sin permiso o sin respeto alguno a los registros de marcas.

Tampoco es nuestra misión aquí, hacer acusaciones concretas ya que los casos que están ante los tribunales son éstos los que deben dictaminar y —lamentablemente— en España se mueven muy despacio y sin medidas provisionales. Y si se refiere a los casos de auténtica copia robada pero por parte de "revistas" en forma de cassette, entonces ya es el caos, y en estos casos los beneficios sí que son astronómicos porque nos metemos en cifras de millones.

Por favor, a ver si hay algún señor legislador que lea esto, conozca lo que es la informática y con qué se come eso del soft y comience a pensar en la necesidad de tomar medidas legales urgentemente o presionar para que se tomen. Y esta denuncia que hacemos no es precisamente porque nos afecte a nosotros que no nos dedicamos a ello, ni tenemos anunciantes de este tipo.

8) A.J. de San Sebastián nos dice que no sabe quién es el autor del artículo.

De los editoriales se hacen responsables las revistas, periódicos, etc., en conjunto. En Commodore World, el autor básico, suele ser María López que dirige la revista. Este es el caso tanto del editorial en cuestión como de esta respuesta.

9) Y ya para terminar y antes de que alguno de vosotros se nos eche encima...

Ya sé que según la regla de tres apuntada, nuestro servicio de cintas puede parecer abusivísimo y la revista en disco, sin embargo, bastante decente. Hay una gran diferencia. Los discos no se hacen uno a uno sobre petición, se hace una tirada determinada y se vende o se envía hasta que se agote.

La cinta, sin embargo, se graba cada cual por separado sobre pedido concreto y nos topamos con el problema de compañía. Hay que conectar, desconectar, buscar ese programa concreto, alguien tiene que dejar lo que está haciendo... la cadena... traducido en personal y tiempo de trabajo. Por eso estamos intentando buscar una forma de grabación turbo que no se atasque en el 50% de los cassettes Commodore y poder ofrecer en cinta el mismo servicio que en disco, no programas sueltos, sino la revista completa a un precio razonable. En cuanto resolvamos el problema lo anunciaremos pero nos está dando más quebraderos de cabeza de los que pensábamos originalmente.

Fallo del I Concurso Nacional de Programación Escolar

El sábado día 20, en Informat, se fallaron los premios del I Concurso Nacional de Programación Escolar en la Sala de protocolo de la Feria de Barcelona.

Solamente unas líneas muy rápidas de ultísima hora para incluir la noticia en este número, al cierre.

El próximo mes, daremos la información completa.

Los ganadores han sido:

Categoría FP

Centro de estudios: Instituto de FP "El Vendrell" de Vendrell (Tarragona).
Equipo: Gloria Olive Marti. M.ª Antonia Miravent Pujol (3.er curso de 2º grado de FP, rama Administrativa y Comercio).
Profesor coordinador: Jordi Francesch Rovira.
Programa: "PENINBAS" (programa de simulación de funcionamiento interno del ordenador).
Ordenador utilizado: PC y compatibles.

Categoría BUP

Centro de estudios: Inst. José de Churriguera de Leganés (Madrid).
Equipo: José Miguel Moreno Mulas (3º de BUP).
Profesor coordinador: Juan José Hernández.
Programa: "Lápiz Digital".
Ordenador utilizado: VIC-20 con Superexpander.

Categoría EGB

(Este premio ha quedado desierto por falta de calidad de los programas enviados, por lo que ha sido pasado al segundo finalista de la categoría BUP).
Centro de estudios: Instituto de Bachillerato Cardenal Herrera Oria.
Equipo: DATA 100 formado por Félix Herrera Piña y Juan José García Oterino (3º de BUP).
Profesor coordinador: Javier Zabala-Camarero Núñez.
Programa: "Derivada de una función en un punto".
Ordenador utilizado: C-64 con Simon's Basic.

Debido a un problema técnico de última hora hemos tenido que aplazar para el próximo mes el programa "VIC-20 Multicolor" anunciado anteriormente para este número.

EXCURSION POR LA MICROPROGRAMACION

Rincón del Código Máquina


Por Diego ROMERO

En este número veremos otra rutina de utilidad para aquellos que deseen hacerse sus propios programas en Basic, pero que no quieran estar limitados por la velocidad del intérprete en las rutinas lentas.

Se trata de una rutina en código máquina que permite clasificar por orden alfabético los elementos de una matriz de cadenas de caracteres (strings). La ventaja de hacerlo en código máquina es su velocidad, y para tener idea de lo rápido que clasifica, podéis ver la siguiente tabla comparativa de tiempos:

Elementos	Longitud	Basic	C.M.
10	10	1,3	0,017
100	10	119,2	0,800
100	40	130,5	0,817
1000	10	*****	82,400

(Los tiempos de la tabla son segundos)

***** En la tabla no figura el tiempo de clasificación de mil elementos de diez caracteres de longitud en Basic, ya que una hora después de empezar a ordenar la matriz sólo tenía clasificados los ciento treinta primeros. Si alguien lo desea puede esperar a que el programa termine y me cuenta cuánto tarda en hacerlo.

La rutina puede cargarse con cualquiera de los siguientes métodos:

1.—Con la ayuda de un ensamblador, por ejemplo el Macro Assembler de Commodore, tecleando el listado 1 y ensamblándolo posteriormente.

2.—Con la ayuda de algún monitor de código máquina introduciendo el listado hexadecimal (listado 2).

3.—Tecleando el programa cargador Basic del listado 3.

Una vez que se ha cargado la rutina de código máquina en la memoria, se debe guardar en cinta o disco antes de probarla, y luego teclear el listado 4 que sirve para demostrar el funcionamiento de la rutina.

Con el programa del listado 4 podemos comprobar el tiempo que tarda el programa de código máquina para distintas cantidades de elementos y longitudes de estos. También podemos medir la velocidad del símil de Basic que aparece en el listado 5.

La rutina de clasificación del listado 5 (subrutina 1000) no es la mejor que se puede hacer para clasificar en Basic, tampoco es la más rápida (ni la más lenta), pero es muy sencilla y se puede pasar a código máquina fácilmente, sirviendo en este caso para ilustrar un poco la función de cada parte del programa de código máquina y permite que podamos entender el funcionamiento del programa en lenguaje ensamblador con ayuda de los comentarios del margen derecho del listado.

Para hacer muy sencilla la rutina debemos tener algunas limitaciones, que en este caso son: las matrices de cadenas (de strings) que podemos clasificar deben ser de una sola dimensión, y el programa no comprueba si lo son o no. Si le pedimos que clasifique una matriz definida por un DIM Z$ (100,3), el programa destrozará muchas de las variables almacenadas en memoria, ya que considera que se trata de una matriz sencilla como la definida por DIM Z$ (100).

La segunda limitación importante es que debemos dar a la matriz un elemento más de los que precisemos, ya que para saber cuál es el último elemento al ordenarla comprueba que sea un string nulo "", y si no lo encontrase también destrozaría algunas variables. Aunque podíamos haber diseñado la rutina de modo que clasificase todos los elementos de la matriz sin necesidad de este elemento "añadido", lo he evitado, ya que en el segundo caso si la matriz no estuviese totalmente llena de cadenas podríamos obtener como resultado de la clasificación un montón de cadenas vacías en el lugar de los primeros elementos de la matriz, y todos los elementos ocupados pasarían a estar al final de la matriz.

Un detalle también importante es el modo de llamar a la rutina, se hace con el SYS 828 seguido de una coma y el primer elemento de la matriz a clasificar. Este modo de llamar a la rutina permite clasificar las matrices a partir de un cierto elemento sin alterar el orden de los primeros. Si llamamos a la rutina con:

SYS 828, A$(2)

Clasificará la matriz A$(n) a partir del elemento segundo, dejando intactos A$(0) y A$(1).

Como el espacio es limitado y los listados son muchos, dejo que veáis los programas ya que el mejor modo de aprender es estudiando el funcionamiento de programas realizados por otros.

Un detalle interesante que se quedaba atrás es que todos los

EXCURSION POR LA MICROPROGRAMACION

saltos y direccionamientos absolutos los he evitado para permitir que la rutina pueda relocalizarse en cualquier posición de memoria, por lo que si deseáis podéis colocarlo al final de la RAM y bajar el vector de fin de memoria con los populares POKEs a las posiciones 55 y 56.

Aquellos que quieran usar esta rutina en un VIC-20, deberán modificar solamente las direcciones de las llamadas a las rutinas BUSVAR y BUSCOM, que en el VIC son $D08B y $CEFD respectivamente, en lugar de $B08B y $BEFD que corresponden al C-64.

Si a alguien se le ocurre la "genial idea" de comprobar cuánto tarda el programa Basic en clasificar los ciento treinta elementos de diez caracteres de longitud, verá que no es una hora, es mucho menos. Pero si se le ocurre probar los mil elementos de caracteres de longitud y una hora después de comenzar la clasificación pulsa la tecla Stop, verá haciendo PRINT I el número por el que va la clasificación. La causa de esta diferencia de tiempo en clasificar por orden alfabético unos ciento treinta elementos es que en el caso de los mil elementos, debe hacer más comparaciones, y para mayor desgracia nuestra, el intérprete basic suele hacer la jugada de detener la ejecución del programa durante más tiempo y más frecuentemente cuanto mayor número de variables tenga el programa. Durante el tiempo que se para, se dedica a recuperar los bytes que tiene libres entre las diferentes variables, esto lo hace por medio de una rutina llamada "garbage colection".

Gracias a que nuestra rutina de código máquina no mueve las posiciones de las variables, nunca se llama a esta rutina. El programa de código máquina sólo intercambia los punteros de los elementos de la matriz sin cambiar la posición de las cadenas de caracteres, de este modo no desperdicia "huecos" y no es necesario recuperar los bytes libres.

Si queréis probar "a mano" el tiempo que tarda en ejecutarse la rutina de recolección de basura, podéis hacerlo tecleando PRINT FREE (0), y viendo el tiempo que tarda en aparecer el resultado. Este aparecerá inmediatamente si no tenéis variables, pero si habéis ejecutado el programa que define la matriz a$ con unos mil elementos de diez caracteres y los llena de modo aleatorio, podéis esperar casi medio minuto a que conteste el resultado.

Naturalmente si hacéis pruebas con matrices muy grandes conseguiréis el error "out of memory", ya que la memoria está limitada, sobre todo para los propietarios del VIC-20.

En el próximo número prepararé una rutina de búsqueda en código máquina, de modo que podamos combinarla con ésta para utilizarla en un programa de base de datos casero pero rápido.

```
LISTADO 1                         ; ESTE PROGRAMA SIRVE PARA ORDENAR
                                  ; CADENAS DE CARACTERES DE BASIC
                                  ; HACIENDOLO EN CODIGO MAQUINA.
                                  ;
033C              LONGIT   =     $FF       ; LONGITUD MENOR
033C              BUSCOM   =     $AEFD     ; BUSCA COMA
033C              BUSVAR   =     $B08B     ; BUSCA VARIABLE
033C              ELEME1   =     $6A       ; PRIMER ELEMENTO DE MATRIZ
033C              ELEME2   =     $6C       ; SEGUNDO ELEMENTO
033C              POINT1   =     $6E       ; PUNTERO A PRIMER STRING
033C              POINT2   =     $FC       ; PUNTERO AL SEGUNDO
033C                       *=    828
033C 20 FD AE     START    JSR   BUSCOM    ; BUSCA LA COMA
033F 20 8B B0              JSR   BUSVAR    ; CALCULA DIRECCION DEL
0342 85 6A                 STA   ELEME1    ; PRIMER ELEMENTO DE
0344 84 6B                 STY   ELEME1+1  ; LA MATRIZ.
0346 A0 00        LI1010   LDY   #0        ; ESTO COMPRUEBA SI LA
0348 B1 6A                 LDA   (ELEME1),Y  ; LONGITUD DEL PRIMER
034A F0 7A                 BEQ   FIN       ; STRING ES CERO.
034C 85 6E                 STA   POINT1    ; LA GUARDA SI NO ES CERO
034E C8           LI1020   INY
034F B1 6A                 LDA   (ELEME1),Y
0351 99 6E 00              STA   POINT1,Y
0354 C0 02                 CPY   #2
0356 D0 F6                 BNE   LI1020
0358 A5 6A                 LDA   ELEME1    ; AQUI HACE LO MISMO
035A 85 6C                 STA   ELEME2    ; QUE EL J=I DEL
035C A5 6B                 LDA   ELEME1+1  ; EQUIVALENTE BASIC.
035E 85 6D                 STA   ELEME2+1
0360 18           LI1030   CLC             ; AQUI HACE LO MISMO
0361 A5 6C                 LDA   ELEME2    ; QUE EL J=J+1 DEL BASIC
0363 69 03                 ADC   #3        ; (APUNTA AL SIGUIENTE
0365 85 6C                 STA   ELEME2    ; ELEMENTO DE LA MATRIZ)
0367 90 02                 BCC   LI1040    ; SALTA SI NO HAY ACARREO
0369 E6 6D                 INC   ELEME2+1  ; EN LA SUMA.
```

```
1350:  036B A0 00   LI1040  LDY  #0          ; COMPARA LA LONGITUD
1360:  036D B1 6C           LDA  (ELEME2),Y  ; DEL SEGUNDO ELEMENTO
1370:  036F F0 47           BEQ  LI1070      ; CON EL PRIMERO Y
1380:  0371 85 FC           STA  POINT2      ; MEMORIZA LA MENOR DE
1390:  0373 C5 6E           CMP  POINT1      ; LAS DOS.
1400:  0375 90 02           BCC  MAYOR
1410:  0377 A5 6E           LDA  POINT1
1420:  0379 85 FF   MAYOR   STA  LONGIT
1430:  037B C8      LAZO1   INY
1440:  037C B1 6C           LDA  (ELEME2),Y
1450:  037E 99 FC 00        STA  POINT2,Y
1460:  0381 C0 02           CPY  #2
1470:  0383 D0 F6           BNE  LAZO1
1480:  0385 A0 00           LDY  #0          ; COMPARA CADA LETRA
1490:  0387 B1 6F   LI1050  LDA  (POINT1+1),Y ; DE LOS DOS STRINGS
1500:  0389 D1 FD           CMP  (POINT2+1),Y ; SALTA SI SON IGUALES.
1510:  038B F0 04           BEQ  SIGLET
1520:  038D 90 D1           BCC  LI1030
1530:  038F B0 09           BCS  CAMBIA      ; INTERCAMBIA LOS STRINGS
1540:  0391 C8      SIGLET  INY              ; PASA A LA SIGUIENTE
1550:  0392 C4 FF           CPY  LONGIT      ; LETRA. SALTA SI QUEDAN
1560:  0394 D0 F1           BNE  LI1050      ; MAS LETRAS.
1570:  0396 C4 6E           CPY  POINT1      ; LONGITUD DEL PRIMERO
1580:  0398 B0 C6           BCS  LI1030      ; SALTA SI ES MENOR.
1590:  039A A0 00   CAMBIA  LDY  #0          ; INTERCAMBIA LOS
1600:  039C B1 6A   LAZO2   LDA  (ELEME1),Y  ; HACE EL B$=A$(I)
1610:  039E AA              TAX
1620:  039F B1 6C           LDA  (ELEME2),Y  ; HACE EL EQUIVALENTE
1630:  03A1 91 6A           STA  (ELEME1),Y  ; AL A$(I)=A$(J).
1640:  03A3 99 6E 00        STA  POINT1,Y
1650:  03A6 8A              TXA
1660:  03A7 91 6C           STA  (ELEME2),Y  ; HACE EL A$(J)=B$
1670:  03A9 C8              INY
1680:  03AA C0 03           CPY  #3
1690:  03AC D0 EE           BNE  LAZO2
1700:  03AE A0 00           LDY  #0
1710:  03B0 B1 6A           LDA  (ELEME1),Y
1720:  03B2 85 FF           STA  LONGIT
1730:  03B4 A9 00           LDA  #0
1740:  03B6 F0 A8           BEQ  LI1030      ; SIGUIENTE ELEMENTO (J)
1750:  03B8 18      LI1070  CLC
1760:  03B9 A5 6A           LDA  ELEME1      ; HACE EL EQUIVALENTE
1770:  03BB 69 03           ADC  #3          ; AL I=I+1 DEL BASIC.
1780:  03BD 85 6A           STA  ELEME1
1790:  03BF 90 02           BCC  SINARR      ; SALTA SI NO HAY
1800:  03C1 E6 6B           INC  ELEME1+1    ; ARRASTRE EN LA SUMA.
1810:  03C3 18      SINARR  CLC              ; ESTO HACE QUE SALTE
1820:  03C4 90 80           BCC  LI1010      ; SIEMPRE.
1830:  03C6 60      FIN     RTS
```

LISTADO 2

```
.:033C 20 FD AE 20 8B B0 85 6A
.:0344 84 6B A0 00 B1 6A F0 7A
.:034C 85 6E C8 B1 6A 99 6E 00
.:0354 C0 02 D0 F6 A5 6A 85 6C
.:035C A5 6B 85 6D 18 A5 6C 69
.:0364 03 85 6C 90 02 E6 6D A0
.:036C 00 B1 6C F0 47 85 FC C5
.:0374 6E 90 02 A5 6E 85 FF C8
.:037C B1 6C 99 FC 00 C0 02 D0
.:0384 F6 A0 00 B1 6F D1 FD F0
.:038C 04 90 D1 B0 09 C8 C4 FF
.:0394 D0 F1 C4 6E B0 C6 A0 00
.:039C B1 6A AA B1 6C 91 6A 99
.:03A4 6E 00 8A 91 6C C8 C0 03
.:03AC D0 EE A0 00 B1 6A 85 FF
.:03B4 A9 00 F0 A8 18 A5 6A 69
.:03BC 03 85 6A 90 02 E6 6B 18
.:03C4 90 80 60 00 00 00 00 00
.:03CC 00 00 00 00 00 00 00 00
```

LISTADO 3

```
10 I=828
20 READ A:IF A=256 THEN 40
30 POKE I,A:I=I+1:S=S+A:GOTO 20
40 READ SU:IFS<>SU THEN PRINT
"ERROR[SPC] EN[SPC]DATAS'
50 END
828 DATA 32,253,174,32,139,176,133,1
```

EXCURSION POR LA MICROPROGRAMACION

```
06
836 DATA 132,107,160,0,177,106,240,1
22
844 DATA 133,110,200,177,106,153,110
,0
852 DATA 192,2,208,246,165,106,133,1
08
860 DATA 165,107,133,109,24,165,108,
105
868 DATA 3,133,108,144,2,230,109,160

876 DATA 0,177,108,240,71,133,252,19
7
684 DATA 110,144,2,165,110,133,255,2
00
892 DATA 177,108,153,252,0,192,2,208

900 DATA 246,160,0,177,111,209,253,2
40
908 DATA 4,144,209,176,9,200,196,255

916 DATA 208,241,196,110,176,198,160
,0
924 DATA 177,106,170,177,108,145,106
,153
932 DATA 110,0,138,145,108,200,192,3

940 DATA 208,238,160,0,177,106,133,2
55
948 DATA 169,0,240,168,24,165,106,10
5
956 DATA 3,133,106,144,2,230,107,24
964 DATA 144,128,96,256,18719
```

```
LISTADO 4
100 OPEN4,4
110 INPUT "NUMERO,[SPC]LONGITUD";N,L

120 DIM A$(N):N=N-1
130 FOR I=0 TO N
140 FOR J=1 TO RND(0)*L
150 A$(I)=A$(I)+CHR$(RND(0)*59+32)
160 NEXT:NEXT
```

```
170 FOR I=0 TO N:PRINTA$(I):NEXT
180 PRINT#4:CLOSE4
190 TI$="000000":SYS828,A$(0):T$=TI$

200 OPEN4,4
210 PRINT#4,:PRINT#4,"TIEMPO[SPC]ORD
ENACION[SPC]=[SPC]"TI$"HHMMSS":PRINT
#4,:PRINT#4,
220 FORI=0TON:PRINTA$(I):NEXT
230 PRINT#4:CLOSE4
240 END
```

```
LISTADO 5

100 OPEN4,4
110 INPUT "NUMERO,[SPC]LONGITUD";N,L

120 DIM A$(N):N=N-1
130 FOR I=0 TO N
140 FOR J=1 TO RND(0)*L
150 A$(I)=A$(I)+CHR$(RND(0)*59+32)
160 NEXT:NEXT
170 FOR I=0 TO N:PRINT#4,A$(I):NEXT
180 PRINT#4:CLOSE4
190 TI$="000000":GOSUB1000:T$=TI$
200 OPEN4,4
210 PRINT#4,:PRINT#4,"TIEMPO[SPC]ORD
ENACION[SPC]=[SPC]"T$"HH.MM.SS.":PRI
NT#4,:PRINT#4,
220 FORI=0TON:PRINT#4,A$(I):NEXT
230 PRINT#4:CLOSE4
240 END
1000 I=0
1010 IFA$(I)=""THENRETURN
1020 J=I
1030 J=J+1
1040 IFA$(J)=""THEN1070
1050 IFA$(I)>A$(J)THENB$=A$(I):A$(I)
=A$(J):A$(J)=B$
1060 GOTO1030
1070 I=I+1:GOTO1010
```

Clave para interpretar los listados

Todos los listados que se publican en esta Revista han sido ejecutados en el modelo correspondiente de la gama de ordenadores COMMODORE. Para facilitar la edición de los mismos en la Revista y para mejorar su legibilidad por parte del usuario, se les ha sometido a ciertas modificaciones mediante un programa escrito especialmente para ello. Para los programas destinados a los ordenadores VIC-20 y COMMODORE 64, en los que se usan frecuentemente las posibilidades gráficas que aparecen del teclado, se han sustituido los símbolos gráficos que aparecen normalmente en los listados por una serie de letras entre corchetes [] que indican la secuencia de teclas que se deben pulsar para obtener el carácter deseado. A continuación se da una tabla para aclarar la interpretación de las indicaciones entre corchetes:

[CRSRD]= Tecla cursor hacia abajo (sin SHIFT)
[CRSRU]= Tecla cursor hacia arriba (con SHIFT)
[CRSRR]= Tecla cursor a la derecha (sin SHIFT)
[CRSRL]= Tecla cursor a la izquierda (con SHIFT)
[HOME]= Tecla CLR/HOME (sin SHIFT)
[CLR]= Tecla CLR/HOME (con SHIFT)

Las indicaciones [BLK] a [YEL] corresponden a la pulsación de las teclas de 1 a 8 junto a la tecla CTRL. Lo mismo sucede con [RVSON] y [RVSOF] respecto a la tecla CTRL y las teclas 9 y 10.

El resto de las indicaciones constan de la parte COMM o SHIF seguidas de una letra, número o símbolo —por ejemplo [COMM+] o [SHIFA]—. Esto indica que para obtener el gráfico necesario en el programa deben pulsarse simultáneamente las teclas COMMODORE (la que lleva el logotipo) o una de SHIFT y la tecla indicada por la letra, el número o el símbolo, en el ejemplo anterior: COMMODORE y + o SHIFT y A, respectivamente.

En los signos gráficos además se cuenta el número de veces que aparece. Por ejemplo, [7 CRSRR] equivale a 7 pulsaciones de la tecla cursor a la derecha y [3 SPC] tres pulsaciones de la barra espaciadora. •

Por EA4LI

Interface CW para el C-64 y el VIC-20

En este número publico el más sencillo de los interfaces que permiten utilizar nuestro transceptor de radioaficionado para emitir y recibir en telegrafía (CW).

El esquema eléctrico del interface completo aparece en la figura 1, y los valores de sus componentes aparecen en la lista adjunta.

Este interface, siendo muy sencillo, sólo necesita un pequeño ajuste. Para realizarlo deberemos regular la resistencia ajustable R-6 hasta que, en ausencia de señal de entrada, el oscilador del circuito integrado genere una frecuencia de 850 Hertzios, esta señal podrá medirse con la ayuda de un frecuencímetro (o algo similar) en la patilla número 5 del circuito integrado.

Quien no disponga de un frecuencímetro digital para realizar el ajuste, puede hacerlo conectando el generador de calibrado de su receptor y calibrando el dial, luego puede desintonizarlo unos 800 ó 900 Hz. Para que la frecuencia de batido sirva de generador de 850 Hz., entonces aplicando la señal a la entrada del circuito decodificador deberá variar el ajuste de R-6 hasta conseguir que el diodo LED se ilumine al máximo.

Como ya dije en otro artículo, el circuito integrado LM-567 no es más que un simple detector de tonos, pero permite recibir tanto telegrafía como radio-teletipo. Este último se puede recibir sintonizando el dial de modo que uno de los dos tonos (marca o espacio) nos dé una señal de 850 Hz. Deberemos sintonizar sobre la marca o sobre el espacio según la estación esté transmitiendo en los modos normal o inverso.

Por tanto, para recibir correctamente una estación, debemos poner el mando selector de modalidad de recepción en CW o SSB, y mover el dial hasta recibir la señal deseada con una frecuencia de batido respecto a nuestro oscilador de unos 800 Hz. Si se trata de telegrafía (CW), no demos tener problemas, ya que el programa publicado adapta su velocidad de trabajo según la cadencia de puntos y rayas recibidas. Pero si deseamos recibir estaciones de radioteletipo tendremos que averiguar, a qué velocidad transmite (generalmente entre 45 y 75 Baudios), y en qué modo lo hace (normal o inverso), para ello sólo existe una solución... ¡Paciencia! y retocar muy despacio la sintonía mientras variamos la velocidad en el programa.

La parte de emisión no necesita ningún ajuste, ya que es un simple transistor que actúa como interruptor en paralelo con el manipulador, pero esto es válido sólo para equipos transistorizados que trabajen con una pequeña tensión positiva en el borne activo del manipulador, y su borne pasivo es tierra. Si nos encontramos frente a un circuito que utiliza tensión negativa o que es alta, es preferible conectar un relé al colector del transistor MC-140, y utilizar los contactos del relé para manipular el equipo emisor. En caso de dudas siempre es preferible utilizar el relé.

Como originalmente este circuito está destinado solamente a la emisión y recepción de telegrafía, no puede utilizarse para emitir en radioteletipo. Hasta el momento no he tenido tiempo de preparar el esquema del modem bueno para RTTY, si durante este mes tuviese tiempo, apareceria en el próximo número.

UNA ADVERTENCIA IMPORTANTE

Ni la revista ni el autor se responsabilizan de cualquier daño que pudiese ser causado al ordenador o emisora por causa de una manipulación o conexión indebida)

Mucho cuidado, pues un simple error en las conexiones o una deficiente toma de tierra en la emisora o una tensión excesiva pueden dañar permanentemente tanto al ordenador como a la emisora, por tanto no es recomendable que ni éste ni ningún otro circuito sea montado o conectado por personas que no tengan el suficiente nivel técnico para realizarlo.

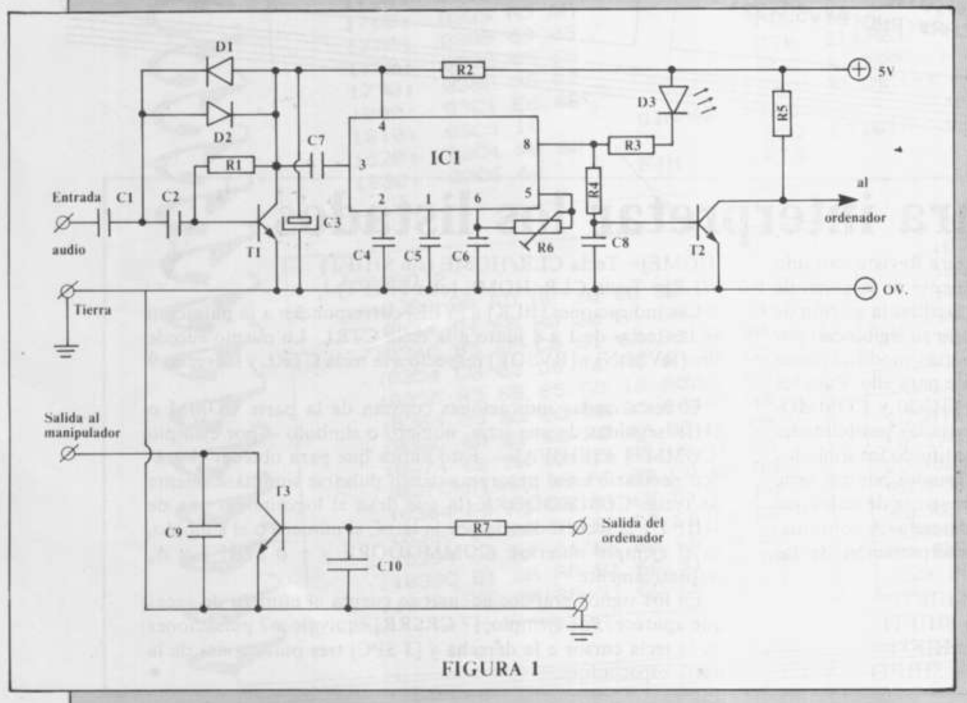


FIGURA 1

Lista de componentes:

RESISTENCIAS
R—1..10 K
R—2..56
R—3..680
R—4..10 K
R—5..1 K
R—6..20 K.ajustable
R—7..15 K

CONDENSADORES
C—1: 22 K
C—2: 22 K
C—3: 100 uF/16 V
C—4: 100 K
C—5: 470 K
C—6: 100 K
C—7: 22 K
C—8: 10 K
C—9: 100 K
C—10: 10 K

SEMICONDUCTORES
D—1: Diodo 1N4148
D—2: Diodo 1N4148
D—3: Diodo LED rojo
T—1: BC—107—B
T—2: BC—107—B
T—3: MC—140
IC—1: LM—567 o equivalente

Todas las resistencias están indicadas en Ohmios, y los condensadores en picofaradios, por tanto una resistencia de 10 K será de 10.000 Ohmios, y un condensador de 10 K será de 10 nanofaradios.